



Gömülü Linux Sistemleri Eğitimi

Kavramlar ve Uygulamalar
05.07.14

Hazırlayan: Nazım KOÇ
<http://ucanlinux.com>



Copyright (C) 2014, Nazım KOÇ

Bu belgenin bütün telif hakları Nazım KOÇ'a aittir.

Bu belgenin tamamı veya bir kısmı,
“kaynak gösterildiği ve değişiklik yapılmadığı” takdirde,
herhangi bir izne gerek kalmadan,
her türlü ortamda çoğaltılabilir, dağıtılabilir, kullanılabilir.
Belgenin son hali her zaman

<http://ucanlinux.com>

adresinden elde edilebilir.

Son güncelleme: 05.07.14

Gömülü Linux

Hazırlayan: Nâzım KOÇ

E-posta : nazim@ucanlinux.com

nazim.koc@gmail.com

Web : <http://ucanlinux.com>

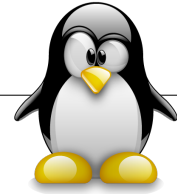


1

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Giriş



➤ Amacımız,

PowerOn ile Login arasında,
uygulanan her adımı
FARKINDA OLARAK yapmaktır.

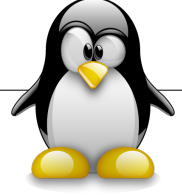
- Bütün çalışmalar Linux'un güzel yüzü "terminal"lerden veya "kara ekran"lardan yapılacaktır.
- Bütün işler el yordamı ile yapılacaktır. Linux'e ait otomatik hiç bir özellik kullanılmayacaktır. Automount, paket yöneticisi, betik vs gibi.
- Çalışmalar Linux dağıtımlarından tamamen bağımsızdır, fakat Ubuntu dağıtımının kullanılması tavsiye edilir.
- Bütün çalışmaların önce teorik kısmı verilecek sonra uygulaması yapılacaktır.

2

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Gömülü Sistem Nedir?



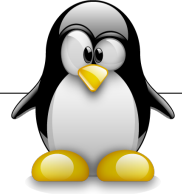
- > Genelde "tek bir işi" çok iyi yapabilmesi için kurulmuş sistemlerdir.
- > Gömülü sistem mevcut bir proje için kurulur. Her işi yapabilen bir gömülü sistem mevcut değildir.
- > Gömülü sistemler proje bazlı kurulduğu için power-on'dan login'e kadar arada geçen bütün aşamalar gömülü sistemi kuran kişi tarafından tek tek yapılmalıdır.
- > Genel amaç için kurulmuş bir sistem alınıp test çalışmaları için kullanılabilir. Fakat müşteriye gidecek son sistem mutlaka projeye özel üretilmelidir ki daha az ram, daha az disk, daha hızlı açılış, daha ufak kök dosya sistemi, daha kolay güncelleme vs yapılabilsin.

3

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Gömülü Sistemlerin Parçaları



Bütün gömülü sistemler genel 4 kısımdan oluşur.

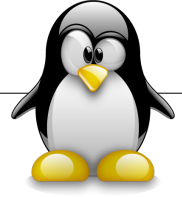
- > Boot loader
Genelde 1. ve 2. seviye açılış yükleyicisi olarak bilinir. (First and second stage boot loader)
1. yükleyici Rom Boot Loader tarafından yüklenir.
2. yükleyici 1. yükleyici tarafından yüklenir.
- > Kernel
Linux çekirdeğidir.
Kaynak kodundan, projeye göre derlenir.
Genelde 2. yükleyici tarafından yüklenir.
- > Root File System
Çekirdek tarafından bağlanan ilk dosya sistemidir.
- > Bootup Scripts
Kök dosya sisteminde çalışır, sistemi ayağa kaldırır.

4

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Gömülü Sistemlerin Parçaları



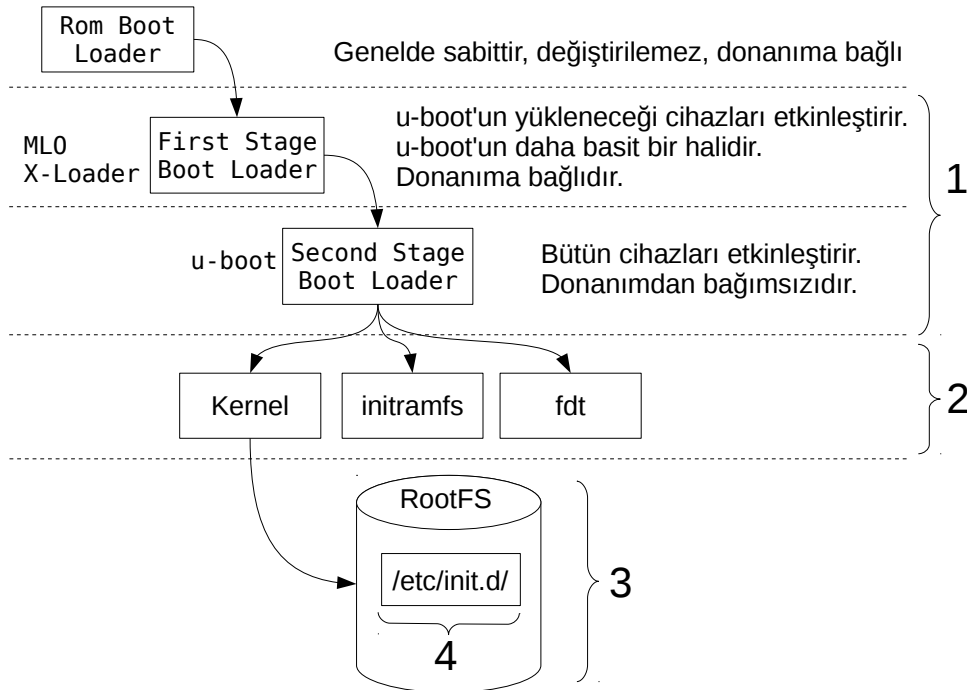
- Burada bahsedilen kısımların bazıları olmayabilir veya arada başka kısımlar bulunabilir.
- Örneğin "intermedite rootfs" in eklenmesi bu yapıyı bozar. initramfs, kernel ve rootfs arasında bir yeredir.
- u-boot bulunmak zorunda değildir. Çekirdek parametreleri doğrudan çekirdek derlenirken verilir ve uygun bir rom kodu ile çekirdek yüklenebilir.
- rootfs olmak zorunda değildir, raw fs derlenmiş bir C programı doğrudan yürütülebilir.
- Açılış betikleri olmayabilir. Örneğin doğrudan shell başlatılabilir. init=/bin/bash gibi.
- Fakat bu ve bunun gibi güncellemeler 4 temel unsurun genelliğini bozmazlar. Sadece özel durum teşkil ederler.

5

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Gömülü Sistemlerin Parçaları, örnek bir açılış



6

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Sistem Yönetimi ve Güncellenmesi



Gömülü sistemler tasarlanırken aşağıdaki gibi verilen bazı işlemlerin gözönünde bulundurulması gerekir.

- > Kernel veya RootFS hatalarında mutlaka acil bir kurtarma sistemi mevcut olmalıdır.

Bunun için genelde 2 adet çekirdek sistemde bulundurulur. u-boot sıra ile birinden açmayı dener. Gerekirse u-boot çekirdek yükleme yapabilir. Yedek ve ufak bir RootFS bulunudurulabilir. initramfs sistemi bu iş için kullanılabilir. Yedek bir RootFS tutulabilir, vs.

- > Uzaktan yönetim gerekli olacak mıdır?
- > Güvenlik nasıl sağlanacaktır?
Kurcalanmaya(temper) karşı önlem alınmış mıdır?
u-boot, env, çekirdek, rootfs ve uygulamaların güvenliği nasıl sağlanacaktır? Secure boot chain? tpm? ...

7

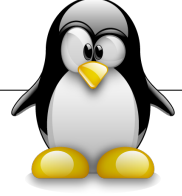
Sistem Yönetimi ve Güncellenmesi



- > Uzaktan güncelleme.
Genelde u-boot güncellenmez.
Fakat çekirdek, RootFS ve uygulamalar güncellenebilir.
Çekirdek güncelleme en basit olanıdır.
Fakat RootFS, yapısı itibari ile dinamik kütüphane bulundurur ve dinamik kütüphanelerin güncellenmesi çok dikkat ister.
Genelde 2 RootFS tutulur.
Güncelleme hatalı biterse, diğer RootFS'te açılış yapılır.
Her zaman uygulama ve RootFS ayrı disk bölümlerinde olmalıdır.
RootFS her zaman RO bağlanmalıdır.
Güncelleme için güvenli bir iletişim sistemi kullanılmalıdır, vs.

- > ...

8



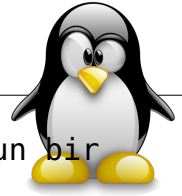
İdeal Test Ortamı

9

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Test Ortamı

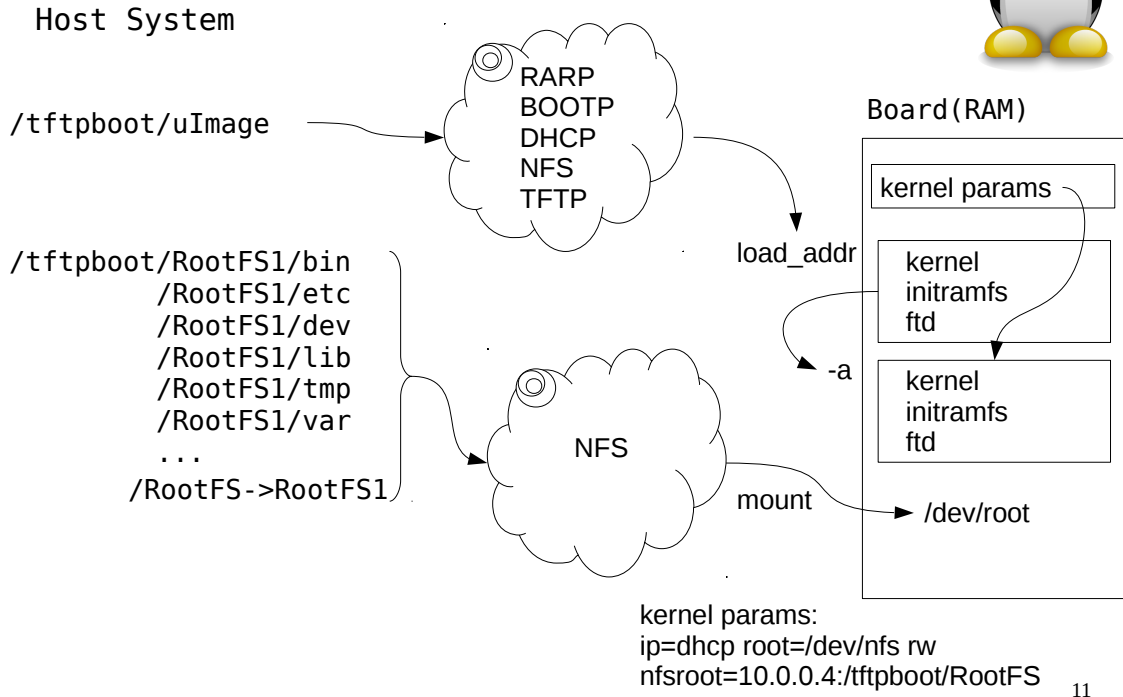


- › Gömülü Linux projesine başlamadan önce mutlaka uygun bir test ortamı kurulmalıdır.
- › En ideal test ortamı NFS üzerinden kurulandır.
- › Bu test ortamında bord üzerinde u-boot'un çalışması yeterlidir.
- › Bu sayede bord üzerindeki MMC, NAND, NOR vs gibi diskler kullanılmayacak her güncelleme işinden sonra bu cihazlara yazım yapmaya gerek kalmayacaktır.
- › Aynı zamanda birden fazla bord ile aynı anda test yapılabilir.
- › Özellikle NAND disklerle hiç çalışmaya gerek kalmadan NFS üzerinden testler yapılabilir, bu da geliştirme hızını inanılmaz ölçüde artıracaktır.

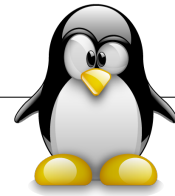
10

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Test Ortamı

Gömülü Linux eğitimleri, kuruluşu ve desteği

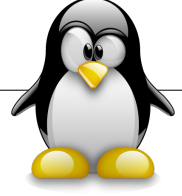
<http://ucanlinux.com>*Test Ortamı*

- Verilen şekilde ok yönlerinin önemi yoktur.
- BORD ağa bağlanabilmelidir.
- Kullanılan u-boot sisteminin ethernet desteği olmalıdır.
- Güvenlikten dolayı bu test ortamının kapalı bir ağda olması tavsiye edilir.
- Host tarafında tftp, dhcp gibi sunucuların çalışır ve düzgün ayarlandığı kabul edilmektedir.
- Basit bir sembolik link ile aynı anda pek çok kök dosya sistemi test edilebilir.
Doğrudan NAND veya MMC ile bu tür testlerin yapılması çok zaman alacaktır.
- minicom, telnet, ssh vs ile terminal girişi elde edilebilir.

12

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



Dosya Sistemleri

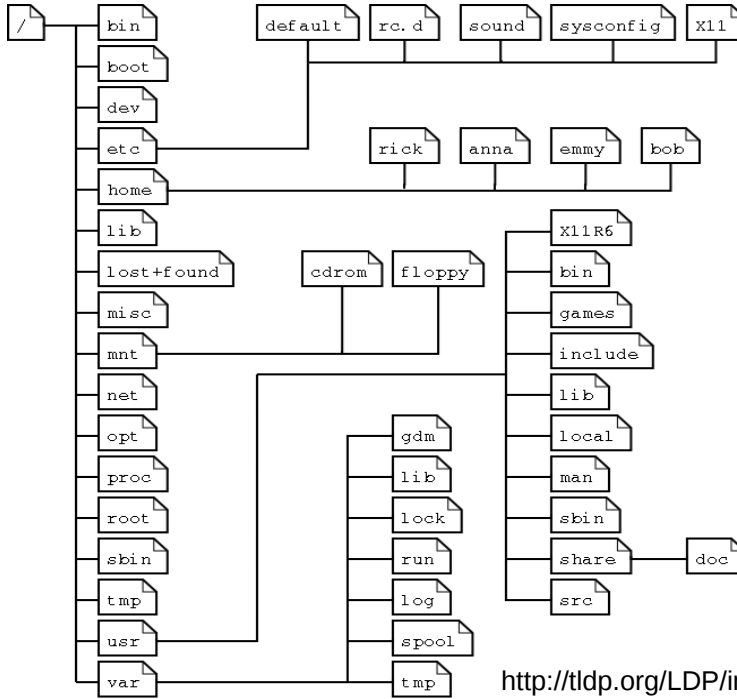
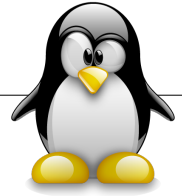
Projeye uygun dosya sistemini seçmek için mevcut dosya sistemlerinin özelliklerinin bilinmesi gerekir?

13

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Dosya Sistemleri



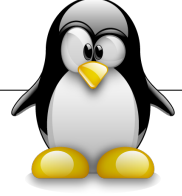
http://tldp.org/LDP/intro-linux/html/sect_03_01.html

14

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Dosya Sistemleri



> İncelenecek Dosya Sistemleri

- 1.vfat
- 2.ext2
- 3.ext3
- 4.ext4
- 5.tmpfs
- 6.ramdisk
- 7.initramfs
- 8.cpio
- 9.NFS
- 10.cramfs
- 11.romfs
- 12.squashfs
- 13.zram
- 14.eCryptfs
- 15.ubifs
- 16.pseudo&virtual fs

15

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Dosya Sistemleri



- > Dosya ve dizinlerin saklandığı yapılardır.
- > Dosya sistemleri ve parametreleri amaca uygun seçilmelidir.
- > Kötü seçim
kaynak israfına
fiziksel yapının bozulmasına
sebebe olabilir.
- > Linux pek çok dosya sistemine destek verir.
- > Çok az dosya sistemi Gömülü Sistemler için uygundur.
- > Dosya sistemi öncelikle cihaza uygun seçilmelidir.
- > Prensip olarak swap kullanılmamalıdır.

16

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Dosya Sistemleri



Dosya sistemleri genelde 5 ortamda kurulur.

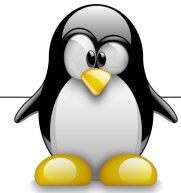
1. SD/MMC Card
vfat, ext2, ext3, ext4,...
2. NAND/NOR Flash
ubifs, jffs2, yaffs2,...
3. RAM
ramdisk, tmpfs, ramfs, zramfs, ...
4. Network
NFS, samba, ...
5. Linux Kernel (Pseudo- and virtual)
/dev, /proc, /sys, /dev/pts, /debugfs, ...

17

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Dosya Sistemleri



- > Seçilen her dosya sistemi için çekirdeğe uygun destek verilmelidir.
- > Dosya sistemleri genelde çekirdekte çok yer kaplarlar. Kullanılmayan dosya sistemleri çekirdeğe eklenmemelidir.
- > `cat /proc/filesystems` komutu ile çekirdeğin o an için desteklediği dosya sistemleri incelenebilir.
- > Kök dosya sistemi hariç, bütün dosya sistemleri modül olarak derlenebilir.
- > Çekirdeğin bağıladığı ilk dosya sistemine kök dosya sistemi denir.

18

Gömülü Linux eğitimleri, kuruluşu ve desteği

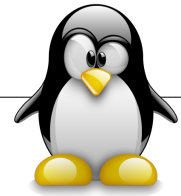
<http://ucanlinux.com>

vfat

- > MS-Windows dosya sistemidir.
- > 255 karaktere kadar dosya isimlerine izin verir.
- > Linux ihtiyaçlarını karşılayamaz.
Sahiplik, mod, sembolik link vs. mevcut değildir.
Asla kök dosya sistemi vfat olmamalıdır.
- > İç yapısı aşırı basit olduğu için daha çok u-boot gibi
açılış-yükleyicileri tarafından kullanılır.
- > Genelde SD/MMC kartların 1. bölümüne kurulur.

19

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*vfat*

```
# cd /tmp

# dd if=/dev/zero of=disk1.img bs=1M count=32
32+0 records in
32+0 records out
33554432 bytes (34 MB) copied, 0,0855615 s, 392 MB/s

# mkfs.vfat disk1.img
mkfs.vfat 3.0.14 (23 Jan 2023)

# mkdir /mnt/disk1

# mount disk1.img /mnt/disk1

root@nkoc:/tmp# df /mnt/disk1
Filesystem      1K-blocks  Used Available Use% Mounted on
/dev/loop0      32686      0    32686    0% /mnt/disk1

# ls -l /mnt/disk1
total 0
```

20

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

vfat

```
# mkfs.vfat /dev/mmcblk0p1
mkfs.vfat 3.0.14 (23 Jan 2023)

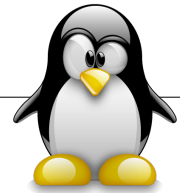
# mkdir /mnt/disk1

# mount disk1.img /mnt/disk1

root@nkoc:/tmp# df /mnt/disk1
Filesystem      1K-blocks  Used Available Use% Mounted on
/dev/loop0           32686     0    32686   0% /mnt/disk1

# ls -l /mnt/disk1
total 0
```

21

vfat

Kernel Support:

```
File systems -->
DOS/FAT/NT Filesystems -->
<*> VFAT (Windows-95) fs support
```

22

ext2

- › Linux'un en eski dosya sistemlerinden biridir.
- › Çok uzun zamandan beri ext2 ile ilgili hiç bir "bug report" yapılmamıştır.
- › Kök dosya sistemi için gerekli bütün ihtiyaçları karşılar.
- › Masaüstü sistemlerde artık kullanılmamaktadır.
- › Ani kapanmalara karşı çok dayanıksızdır.
- › MMC cihazlarında, R0 modu ile bağlanarak kullanılmalıdır. Kök dosya sistemi olarak, RW modunda bağlanmamalıdır.
- › Kök dosya sistemi RW modunda kullanılacaksa, önce ro bağlanmalı, sonra fsck yapılmalı, gerekirse repair edilmeli ve sonra rw bağlanmalıdır.

23

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ext2*

```
# mkdir /mnt/disk2

# dd if=/dev/zero of=disk2.img bs=1M count=32
32+0 records in
32+0 records out
33554432 bytes (34 MB) copied, 0,069849 s, 480 MB/s

# mkfs.ext2 disk2.img

# mount disk2.img /mnt/disk2

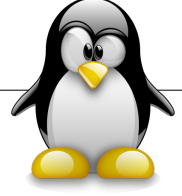
# df /mnt/disk2
Filesystem      1K-blocks  Used Available Use% Mounted on
/dev/loop1      31729     395    29696    2% /mnt/disk2
```

24

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ext2



Kernel Support:

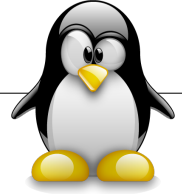
File systems -->
<*> Second extended fs support

25

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ext3



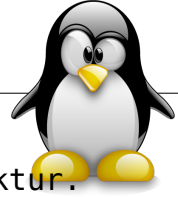
- > ext2'nin devamı ve log tabanlıdır.
- > Ani kapanmalara karşı çok dayanıklıdır. Dosya bütünlüğü çok zor bozulur.
- > Recovery durumu çok hızlıdır.
- > Yerini ext4 sistemine bırakmıştır. Masaüstü sistemlerde artık ext4 kullanılmaktadır.
- > MMC cihazlarında, RW modu ile bağlanarak kullanılabilir.
- > Log tabanlı bir sistem olduğu için, diske yazılacak bütün veriler önce bir log tablosunda tutulur, daha sonra diske yazılır.

26

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ext3



- > Crash durumunda sonra e2fsck çalıştırmaya gerek yoktur. Log sayesinde dosya bütünlüğü her zaman sağlanır.
- > ext3 = journal + ext2
- > ext2 ve ext3 arasında her zaman geçiş yapılabilir.
- > Performans ve dosya bütünlüğü arasında ters ilişki vardır.

data=write_back: crash sonrası yazılmayan veri kalabilir.
data=ordered: Daha güvenlidir ama yavaştır. Tavsiye edilir.

27

ext3

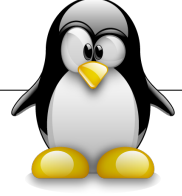


```
# dd if=/dev/zero of=disk3.img bs=1M count=32
# mkfs.ext3 disk3.img
# mkdir /mnt/disk3
# mount disk3.img /mnt/disk3

# df /mnt/disk3
Filesystem      1K-blocks  Used Available Use% Mounted on
/dev/loop2      31729    4508    25583   15% /mnt/disk3
```

28

ext3



Kernel Support:

```
File systems -->
<*> Ext3 journalling file system support
[*]   Default to 'data=ordered' in ext3
```

29

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ext4

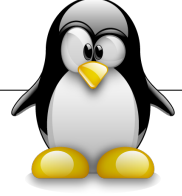


- > ext3'ün devamıdır ama ext3 ile uyumlu değildir. ext4 dosya sistemi, ext3'ü mount edebilir ve ext3'e göre daha iyi performans gösterir.
- > Doğrudan ext4 kullanılması tavsiye edilir.
- > ext3, ext2 ile uyumlu idi. Her iki yönde de geçiş yapılabiliyordu.
- > Örnek dosya sistemi kuruluşu ve çekirdek desteği ext2/ext3 gibidir.
- > Gömülü sistemlerde, MMC tarafında kök dosya sistemi olarak rw modunda mount edilebilir.
- > ext2 sisteminin rw modunda bağlanması tavsiye edilmez.

30

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

tmpfs

- > Sanal bellekte kurulan bir dosya sistemidir.
- > virtual memory = ram + swap
- > Doldukça büyür, içi boşaldıkça küçülürler. Ayrılan kapasiteyi RAM'dan çalmazlar.
- > Swap edilebilirler.
- > Disk cache sisteminin değiştirilmiş bir hali gibidir. Çok az yer kaplar. Bundan dolayı seçilmese bile çekirdek tarafında desteği vardır.
- > /dev/shm, /dev gibi pek çok cihaz, bu dosya sistemini kullanır.

31

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*tmpfs*

- > Kapasite verilmezse, RAM'in yarısı kabul edilir. Çalışma anında kapasite artırılabilir.
- > umount edildiği an veriler kaybolur.
- > Güç kesildiği an veriler kaybolur.

32

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

tmpfs

```
# mkdir /mnt/disk4
```

```
# mount -t tmpfs tmpfs /mnt/disk4
```

```
# df /mnt/disk4
```

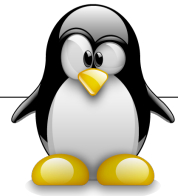
Filesystem	1K-blocks	Used	Available	Use%	Mounted on
tmpfs	2011344	0	2011344	0%	/mnt/disk4

```
root@nkoc:/tmp# free
```

	total	used	free	shared	buffers	cached
Mem:	4022692	2092160	1930532	0	599184	892832
-/+ buffers/cache:		600144	3422548			
Swap:	2097148	0	2097148			

33

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*tmpfs*

Kernel Support:

File systems -->

Pseudo filesystems -->

[*] Virtual memory file system support (former shm fs)

34

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

tmpfs

- > mount edilen bütün dosyalar, işleri bitince umount edilmelidir.
- > 32MB için, ext3 dosya sisteminin maliyeti %15'tir.

```
# df /mnt/disk?
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/loop0	32686	0	32686	0%	/mnt/disk1
/dev/loop1	31729	395	29696	2%	/mnt/disk2
/dev/loop2	31729	4508	25583	15%	/mnt/disk3
tmpfs	2011344	0	2011344	0%	/mnt/disk4

35

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*tmpfs*

```
# mount | grep disk
/tmp/disk1.img on /mnt/disk1 type vfat (rw)
/tmp/disk2.img on /mnt/disk2 type ext2 (rw)
/tmp/disk3.img on /mnt/disk3 type ext3 (rw)
tmpfs on /mnt/disk4 type tmpfs (rw)
```

```
$ umount /mnt/disk1
$ umount /mnt/disk2
$ umount /mnt/disk3
$ umount /mnt/disk4
```

36

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ramdisk



- › RAM'ın (virtual memory değil!) disk bölümü gibi kullanılmasıdır.
- › Dosya sistemi değildir, dosya sistemleri için altyapı hazırlar.
- › Sanki bir disk bölümü (disc partition) gibi üzerine hemen dosya sistemi kurulabilir.
- › tmpfs gibi umount edildiğinde yok olmaz.
- › Cihaz isimleri /dev/ram0, /dev/ram1, ... şeklindedir.
- › Ramdisk adedi, kapasitesi ve blok boyu çekirdek parametresi olarak verilebilir.
- › Varsayılan blok boyu 1024'tür.

37

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ramdisk



- › Mümkünse ramdisk kullanılmamalı, bunun yerine tmpfs kullanılmalıdır.
- › ramdisk, kullanıldığı an RAM bellekten kapasite çalar. umount edilse bile kapasite geri verilmez.
- › Sanal dosya sistemi kullanmadığı için ihtiyaç durumunda swap edilemez.
Prensip olarak Gömülü sistemlerde swap kullanılmaz.
- › Bazı çekirdekler hala initial ram disk kullanmaktadırlar. Fakat initrd kavramı yerini initramfs'e (tmpfs'e) bırakmıştır.

38

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ramdisk

- > tmpfs bir dosya sistemidir. Çekirdek tarafından bizim hiç ilgilenmediğimiz bir yerlerde kurulur!
- > Ramfs veya /dev/ram0 veya ramdisk fiziksel bir ortamdır. Bu ortamı kullanabilmek için illa ki üzerine bir dosya sistemi kurulmalıdır, ext2, ext3, minix, vfat, vs gibi.
- > /dev/ram0 üzerine dosya sistemi kurulur ve mount edilir.
- > tmpfs doğrudan mount edilir.
- > dmesg | grep RAMDISK

39

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ramdisk*

- > 4M'lık ext2 kur ve paketle?
- ```
dd if=/dev/zero of=/dev/ram0 bs=1k count=4096
mkfs.ext2 /dev/ram0 4096
mkdir /mnt/disk
mount /dev/ram0 /mnt/disk
df
mount
umount /mnt/disk

dd if=/dev/ram0 of=/tmp/ram0.img bs=1k count=4096
dd if=/dev/ram0 bs=1k count=4096 | gzip > /tmp/ram0.img.gz
```
- > Doğrudan imaj olarak mount etmek, /dev/ram0 kullanmadan.
- ```
# mount -o loop /tmp/ram0.img /mnt/disk
```

40

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ramdisk



Kernel Support:

```
Device Drivers -->
Block devices -->
  <*> RAM block device support
  (16) Default number of RAM disks
  (65536) Default RAM disk size (kbytes)
```

41

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

initramfs



- > Çekirdek tmpfs destekli bir dosya sistemini kök dosya sistemi olarak bağlayabilir.
- > Bu dosya sistemine initramfs denir, temeli tmpfs dosya sistemidir.
- > Temeli ramdisk olan dosya sistemine ise initrd denir ve artık kullanılmamaktadır.
- > initramfs sistemi çekirdek yüklendikten hemen sonra mount edilir.
- > Genelde modül yüklemek içindir. Fakat gömülü sistemlerde esas kök dosya sistemi olarak da kullanılabilir.
- > Bu dosya sistemi çekirdeğin içine gömülü olabilir.

42

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

initramfs

- > Ayrı bir dosya olarak da bulunabilir.
- > Ayrık olan dosya boot loader tarafından çekirdekle birlikte yüklenir ve başlangıç adresi ve dosya boyu çekirdek parametresi olarak çekirdeğe gönderilir.
- > Ayrık dosya her zaman cpio arşivi şeklindedir, dosya sistemi değildir.
- > Çekirdek kendini açar ve cpio arşivini, bir tmpfs dosya sistemi içine açar.
- > Bu dosya sistemine initramfs denir ve çekirdek bu dosya sistemini kök dosya sistemi olarak bağlar ve /init komutunu gözü kapalı işletir.
- > initrd durumunda /linuxrc işletilir.

43

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*initramfs, çekirdeğe gömülü*

- > /tmp/RootFS kök dosya sistemi aşağıdaki gibi çekirdek koduna eklenebilir.
- ```
[*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
(/tmp/RootFS) Initramfs source file(s)
(1000) User ID to map to 0 (user root)
(1000) Group ID to map to 0 (group root)
[*] Support initial ramdisks compressed using gzip
```
- > /tmp/RootFS altındaki bütün dosya ve dizinler otomatik olarak cpio arşivi haline getirilir, gzip ile sıkıştırılır ve çekirdek kodunda eklenir.
  - > Eğer RootFS çok büyükse bu teknik uygun değildir.
  - > RootFS içindeki her program GPL veya benzeri olmak zorundadır.

44

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

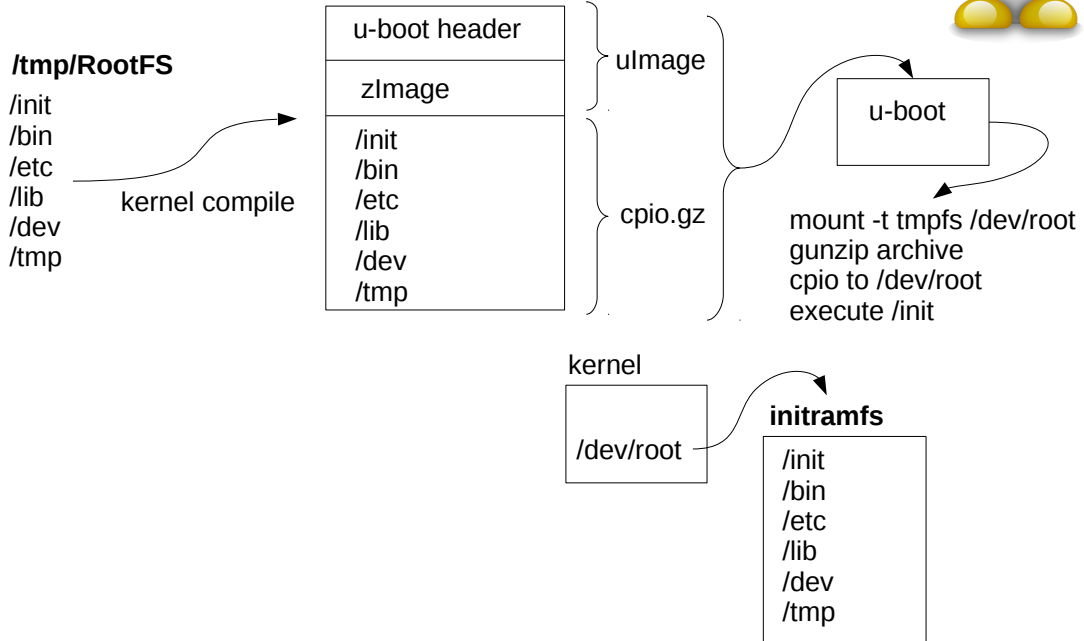
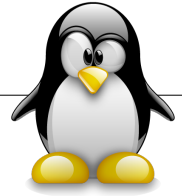


*initramfs, çekirdeğe gömülü*

- Açılış çok hızlıdır, çekirdek yüklendiği an kök dosya sistemi de yüklenmiş gibidir. Aradaki süre çok azdır.
- Masaüstü sistemlerde gerçek kök dosya sistemlerine geçiş için kullanılır. Gömülü sistemlerde acil durum açılışları veya gerçek kök dosya sistemi olarak kullanılabilir.
- RootFS'deki her güncellemede çekirdek yeniden derlenmeli ve sisteme yüklenmelidir.
- Tek bir imaj dosyası ile hem çekirdek hem kök dosya sistemi taşınır ve boot loader sadece tek dosya yükler.

45

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*initramfs, çekirdeğe gömülü*

46

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*initramfs, çekirdekten ayrık*

- › Çekirdek bilinen yolla derlenir.

```
[*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
() Initramfs source file(s)
```

- › /tmp/RootFS için cpio arşivi hazırlanır.

```
find . | cpio -v -o -H newc | gzip > ramfs.gz
mkimage -A arm -T ramdisk -C none
 -n "cpio test imaji" -d ramfs.gz uramfs
```

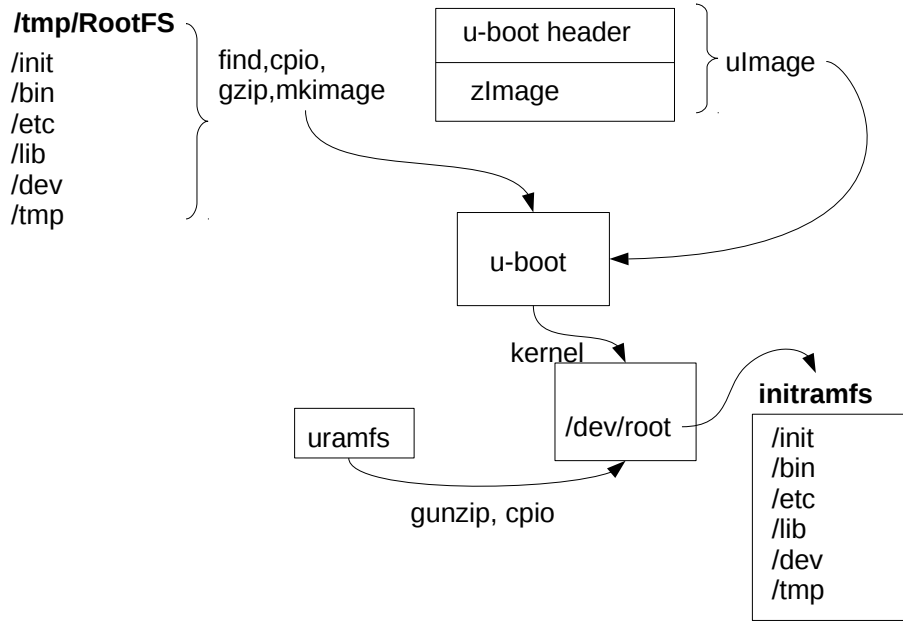
- › uImage ve uramfs beraber kullanılır.
- › uramfs içindeki programların GPL olması gerekmez.
- › Güncellemelerde sadece uramfs kullanılır, çekirdek derlenmez.

47

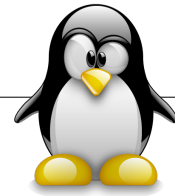
*initramfs, çekirdekten ayrık*

- › u-boot, önce çekirdeği sonra uramfs'i yükler. Çekirdeğe uramfs'in yüklendiği adresi ve boyunu parametre olarak geçirir.
- › Çekirdek, daha önce bahsedildiği gibi arşivi gunzip ile açar, tmpfs içine cpio arşivini kopyalar ve /init programını gözü kapalı başlatır.
- › 2 adet yükleme olduğu için çekirdeğe gömülü sisteme göre biraz daha yavaş açılır.

48

*initramfs, çekirdeğe gömülü*

49

*cpio*

- > cpio bir dosya sistemi değildir, initramfs için arşivleme yapar.
- > cpio kodu çekirdek kodu içinde gömülü durumdadır.
- > Aynı anda ya input ya da output modunda çalışır. output modu paketleme, arşivleme yapar.
- > output modunda dosya isimleri bir dosyadan satır satır okunur.
 

```
cpio -o -H newc < list.txt
ls falan.* | cpio -o -Hnewc > test.cpio
```
- > Çekirdek sadece newc arşiv formatını kullanır.
- > Genelde find ile kullanılır.
 

```
find . | cpio -o -H newc | gzip > rootfs.img.gz
```

50

## *cpio*



> input veya extract modunda paket açılır.

```
gunzip rootfs.img.gz
cpio -i -d -H newc -F rootfs.img --no-absolute-filenames
```

> -i: input veya extract demektir.

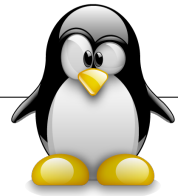
> -o: output veya create demektir.

51

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## *NFS*



> Kök dosya sistemi ağ üzerinden bağlanabilir ya da host üzerindeki herhangi bir dizin export edilebilir.

> /etc/exports:

```
/tftpboot 10.0.0.0/24(rw,insecure,
no_subtree_check,async,no_root_squash)
```

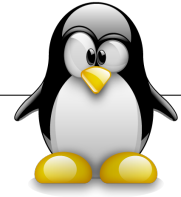
> # exportfs -avr ile yapılan güncellemeler sunucuya bildirilir.

> no\_root\_squash sayesinde root yetkisi ile işlem yapılabilir.  
insecure ile güvenli portların dışında da port numarası kullanılabilir.  
no\_subtree\_check ile dışarı taşan dizinlere erişim sağlanır.

52

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*NFS*

- > Kök dosya sistemi NFS üzerinden kullanılacaksa çekirdek aşağıdaki gibi derlenir.

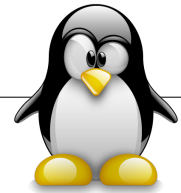
```
[*] Networking support
 Networking options -->
 [*] TCP/IP networking
 [*] IP: kernel level autoconfiguration
```

```
File systems -->
 [*] Network File Systems
 <*> NFS client support
 [*] Root file system on NFS
```

- > Ağ için çekirdek parametreleri

```
ip=<client-ip>:<server-ip>:<gw-ip>:<netmask>:<hostname>:
 <device>:<autoconf>:<dns0-ip>:<dns1-ip>
```

53

*NFS*

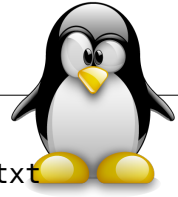
- > Çekirdek parametreleri aşağıdaki gibi olabilir.

```
ip=10.0.0.111:10.0.0.4:10.0.0.4:255.255.255.0:test1:eth0:off
root=/dev/nfs
rw
nfsroot=10.0.0.4:/tftpboot/RootFS
```

- > Sıradan bir dizini mount etmek için,

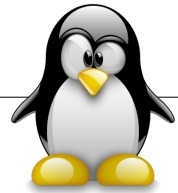
```
mount -t nfs 10.0.0.4:/tftpboot/app /mnt/nfs
...
umount /mnt/nfs
```

54

*NFS*

- > autoconf parameters: Documentation/filesystems/nfsroot.txt
  - off or none: don't use autoconfiguration  
(do static IP assignment instead)
  - on or any: use any protocol available in the kernel  
(default)
  - dhcp: use DHCP
  - bootp: use BOOTP
  - rarp: use RARP
  - both: use both BOOTP and RARP but not DHCP  
(old option kept for backwards compatibility)
- Default: any

55

*cramfs*

- > compressed dosya sistemidir.  
Compressed ROM File System
- > Sadece ro erişim yapılabilir.
- > Dosya boyu 16MB'yı geçemez.
- > FS büyüklüğü 256MB'tan büyük olamaz.
- > Kullanımı sırasında decompress edilmez, her zaman compress durumunda kullanılır.
- > Çok az yer kaplar, gömülü sistemler için uygundur.
- > Mevcut kısıtlamalara dikkat edilerek kullanılmalıdır.  
Not "16/32 bits uid/gid and hard links and timestamps"

56

*cramfs*

```
mkfs.cramfs -v -n test.image /tftpboot/RootFS cramfs.img

ls -l cramfs.img
-rw-rw-r-- 1 nazim nazim 1089536 May 11 10:17 cramfs.img

du -ks /tftpboot/RootFS/
2120 /tftpboot/RootFS/

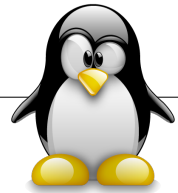
file cramfs.img
cramfs.img: Linux Compressed ROM File System data,
 little endian size 1089536 version #2
 sorted_dirs
 CRC 0x4101b0de, edition 0, 526 blocks,
 388 files

mount -o loop cramfs.img /mnt/disk
ls -l /mnt/disk # timestamp= unix epoch time

df /mnt/disk
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/loop0 2104 2104 0 100% /mnt/disk
```

57

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*cramfs*

Kernel Support:

```
File systems --->
[*] Miscellaneous filesystems --->
 <*> Compressed ROM file system support (cramfs)
```

58

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

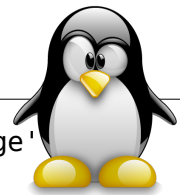
*romfs*

- > Çok az yer kaplar.
- > Sadece ro erişim yapılabilir.

## Kernel Support:

```
File systems --->
[*] Miscellaneous filesystems --->
 <*> ROM file system support
```

59

*romfs*

```
genromfs -d /tftpboot/RootFS -f romfs.img -V 'romfs image'
ls -l romfs.img
-rw-rw-r-- 1 nazim nazim 1977344 May 11 10:40 romfs.img
du -ks /tftpboot/RootFS/
2120 /tftpboot/RootFS/
mount -o loop -o ro -t romfs romfs.img /mnt/disk
df /mnt/disk
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/loop0 1931 1931 0 100% /mnt/disk
ls -l /mnt/disk
cat /proc/filesystems | grep romfs
```

60

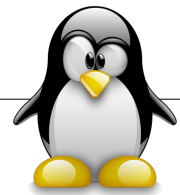


*squashfs*

- > gzip/lzo/xz ile sıkıştırılmış, salt okunur dosya sistemidir.
- > cramfs'e göre daha iyi sıkıştırma yapar.
- > tar.gz'nin mount edilmesi gibi düşünülebilir.
- > Pek çok LiveCD'nin "defacto FS" standardı gibidir.
- > uid/gid, timestamp desteği vardır.  
4GB'tan büyük dosyaları destekler.

61

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*squashfs*

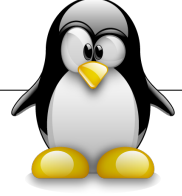
```
mksquashfs /tftpboot/RootFS.busybox rootfs.sq
ls -l rootfs.sq
-rw-r--r-- 1 root root 974848 May 11 12:30 rootfs.sq
du -ks /tftpboot/RootFS.busybox
2120 /tftpboot/RootFS.busybox
file rootfs.sq
mount rootfs.sq /mnt/disk
ls -l /mnt/disk
df /mnt/disk
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/loop0 1024 1024 0 100% /mnt/disk
losetup -a
/dev/loop0: [0801]:655457 (/mnt/rootfs.sq)
unsquashfs rootfs.sq
cd squashfs-root/
```

62

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## *squashfs*



Kernel Support:

```
File systems --->
[*] Miscellaneous filesystems --->
 <*> SquashFS 4.0 - Squashed file system support
```

63

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## *zram*



- Compressed RAM based block device
- Masaüstü sistemlerde swap'ın hızı ve kapasitesini artırmak için kullanılır.
- /dev/ram0'ın sıkıştırılmış hali gözü ile bakılabilir.
- Her zaman modül olarak derlenmesi tavsiye edilmektedir. Böylece disk sayısı amaca göre atanabilir.
- /var, /tmp gibi dizinler için de kullanılabilir. Fakat tmpfs gibi olmayıp doğrudan ram'dan kapasite çalınır.

64

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*zram*

```
modinfo zram
modprobe zram num_devices=4
ls -l /dev/zram?
lsmod|grep zram
echo $((4096*1024)) > /sys/block/zram0/disksize

mkfs.ext2 /dev/zram0
mount /dev/zram0 /mnt/disk

df /mnt/disk
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/zram0 3952 24 3724 1% /mnt/disk

cd /mnt/disk ; ls -la

mount | grep zram
/dev/zram0 on /mnt/disk type ext2 (rw)

dmesg|grep zram
umount /mnt/disk
rmmod zram
lsmod | grep zram
```

65

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*zram*

Kernel Support:

```
Device Drivers --->
 [*] Staging drivers --->
 <M> Compressed RAM block device support
 [] Compressed RAM block device debug support
 [*] Memory allocator for compressed pages
```

66

Gömülü Linux eğitimleri, kuruluşu ve desteği

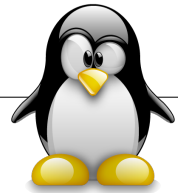
<http://ucanlinux.com>

*eCryptfs*

- > VFS layer üzerinde işler, olgunluğa erişmemiştir.
- > Stack based desteklidir.  
Bundan dolayı mevcut herhangi bir dosya sistemi üzerinde şifreleme yapabilir.
- > Önce boş bir dizin mount edilir.  
Sonra bu dizine dosyalar kopyalanır.  
Diske yazılan dosyalar artık şifrelenmiştir.
- > İçi dolu dizinler mount edilmemelidir.
- > user space tarafında, ecryptfs-utils programlarına gerek vardır.

67

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*eCryptfs*

```
mount -t ecryptfs /srv /srv

more ~/.ecryptfs/sig-cache.txt
1aba6bfb1a2bf34e

echo "Merhaba" > /srv/test
cd /srv/test
ls -l

cat test
Merhaba

df
Filesystem 1K-blocks Used Available Use% Mounted on
/srv 65924860 9332148 53220888 15% /srv

mount | grep ecryptfs
/srv on /srv type ecryptfs (rw,ecryptfs_sig=1aba6bfb1a2bf34e,
ecryptfs_cipher=aes,ecryptfs_key_bytes=16,
ecryptfs_unlink_sigs)
```

68

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*eCryptfs*

```
umount /srv
file /srv/test
cat > ~/.ecryptfsrc
key=passphrase:passphrase_passwd_file=/tmp/passwd_file.txt
ecryptfs_sig=1aba6bfb1a2bf34e
ecryptfs_cipher=aes
ecryptfs_key_bytes=16
ecryptfs_passthrough=n
ecryptfs_enable_filename_crypto=n

cat .ecryptfs/sig-cache.txt
1aba6bfb1a2bf34e

passwd_file.txt, usb, tpm vs gibi bir ortamda olabilir.

cat /tmp/passwd_file.txt
passphrase_passwd=deneme

root@koc:~# mount -t ecryptfs /srv /srv
```

69

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*eCryptfs*

Kernel Support:

```
[*] Cryptographic API --->
 [*] MD5 digest algorithm
 <*> AES cipher algorithms (ARM-asm)

Security options --->
 [*] Enable access key retention support

File systems --->
 [*] Miscellaneous filesystems --->
 <*> eCrypt filesystem layer support (EXPERIMENTAL)
```

70

Gömülü Linux eğitimleri, kuruluşu ve desteği

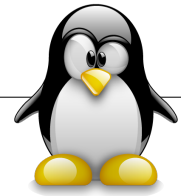
<http://ucanlinux.com>

*ubifs*

- > Flash diskler için geliştirilmiş, kapanmaya dayanıklı ve ubi üzerine kurulan, log tabanlı bir dosya sistemidir.
- > NAND diskler için neredeyse standard gibi olmuştur.
- > Diğer dosya sistemleri gibi doğrudan kurulup mount edilemez, ön hazırlık gerekir.
- > Tamamen boş bir disk mount edilirse otomatik olarak ubifs kurulur, açıkça da kurulabilir.
- > fsck programı yoktur.  
Mount sırasında hata meydana gelirse dosya sistemini düzeltmeye çalışır. Düzeltirse mount eder, düzeltmezse ya ro mount eder ya da hiç mount etmez.
- > loop mount kavramı yoktur, simulator ile mount edilir.

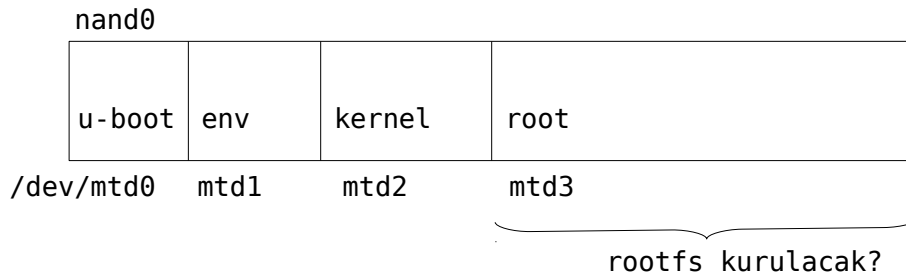
71

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ubifs*

## NAND Partitions

```
mtdparts=nand0:256k@0(u-boot),128k(env),5m(kernel),-(root)
```



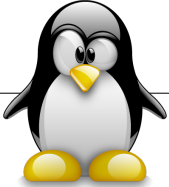
```
/dev/mtd[0123] : raw nand flash
```

72

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*ubifs*



```

mount /dev/ubi3_0
mount /dev/ubi3_1

```

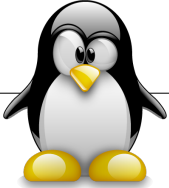
Diagram illustrating the UBI (User Block Image) structure and mounting process:

- Physical Erase Blocks (PEB) are located in `/dev/mtd3` (PEB0, PEB1, PEB2, ..., PEBN).
- Logical Erase Blocks (LEB) are mapped to `/dev/ubi3` (LEB0, LEB1, LEB2, ..., LEBN).
- UBI volumes are created using `ubimkvol /dev/ubi3`.
- UBI volumes are attached to the kernel using `ubiattach -m3 -d3`.
- The UBI volumes are formatted using `ubiformat /dev/mtd3`.
- The UBI volumes are mounted using `mount /dev/ubi3_0` and `mount /dev/ubi3_1`.

73

Gömülü Linux eğitimleri, kuruluşu ve desteği <http://ucanlinux.com>

*ubifs*



```

$ mount -t ubifs ubi0:root /dev/root
$ mount -t ubifs /dev/ubi0_0 /dev/root

```

Diagram illustrating the UBI (User Block Image) structure and mounting process:

- Physical Erase Blocks (PEB) are located in `/dev/mtd3` (PEB0, PEB1, PEB2, ..., PEBN).
- Logical Erase Blocks (LEB) are mapped to `/dev/ubi0` (LEB0, LEB1, LEB2, ..., LEBN).
- UBI volumes are created using `ubimkvol /dev/ubi0 -N root`.
- UBI volumes are attached to the kernel using `ubiattach -m3 -d0`.
- The UBI volumes are formatted using `ubiformat /dev/mtd3`.
- The UBI volumes are mounted using `mount -t ubifs ubi0:root /dev/root` and `mount -t ubifs /dev/ubi0_0 /dev/root`.

74

Gömülü Linux eğitimleri, kuruluşu ve desteği <http://ucanlinux.com>

*ubifs*

- Bord tarafında çalışırken ubifs kuruluşları (1)
- Host tarafında, /tftpboot/RootFS altında örnek bir RootFS olduğunu kabul edelim.

```
$ ubiformat /dev/mtd3
$ ubiattach -p /dev/mtd3
$ ubimkvol /dev/ubi0 -N root -m
$ mount -t ubifs ubi0:root /mnt/root

$ mount -t nfs host:/tftpboot /mnt/nfs
$ cp -a /mnt/nfs/* /mnt/root

$ nanddump /dev/mtd3 -f /mnt/nfs/mtd3.img

$ umount /mnt/root
```

75

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ubifs*

- Bord tarafında çalışırken ubifs kuruluşları (2)
- Host tarafında, ubinize ile root.ubi isimli imajın kurulduğunu kabul edelim.

```
$ ubiformat /dev/mtd3 -f /tftpboot/root.ubi
$ ubiattach -p /dev/mtd3
$ mount -t ubifs ubi0:root /mnt/root
```

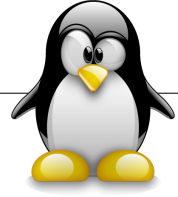
76

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



## ubifs



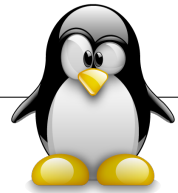
- › Bord tarafında çalışırken ubifs kuruluşları (3)
- › Host tarafında, mkfs.ubifs kurulan imajın yazılması.

```
$ ubiformat /dev/mtd3
$ ubiattach -p /dev/mtd3
$ ubimkvol /dev/ubi0 -N root -m
$ updatevol /dev/ubi0_0 /tftpboot/root.ubifs
$ mount -t ubifs ubi0:root /mnt/root

$ mount -t ubifs /dev/ubi0_0 /mnt/root
```

77

## ubifs



- › Bord tarafında çalışırken ubifs kuruluşları (4)
- › Host tarafında, ubinize ile root.ubi kurulmuş olsun.

U-Boot seviyesinde iken:

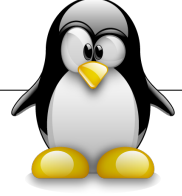
```
> tftp 32000000 root.ubi
> nand erase root
> nand write root.ubi
```

Açıldıktan sonra:

```
$ mount -t ubifs ubi0:root /mnt/root # veya
$ mount -t ubifs /dev/ubi0_0 /mnt/root
```

78

## *ubifs*



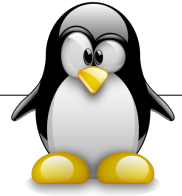
- › Host tarafında çalışırken ubifs kuruluđu:
- › Bord'a ait bir imajı host tarafında oluđuurma işine ubinize denir.
- › Bord tarafında ubinize yapmanın bir anlamı yoktur.
- › Burada esas amaç NAND için master bir imaj oluđuurmaktır.
- › Genelde müşteriye verilecek en son sistem için hazırlanır.
- › Test veya geliştirme için ubinize kullanmak pek pratik değildir.

79

Gömülü Linux eğitimleri, kuruluđu ve desteđi

<http://ucanlinux.com>

## *ubifs*



- › ubinize işlemine önce ubifs kurularak başlanır. Burada NAND imajı ile ilgili bilgi bulunmaz.
- › Bir imaj içinde çok fazla volume veya ubifs bulunabilir. Her bir ubifs'in parametreleri root.ini gibi bir dosyaya yazılır.
- › Çok fazla volume olacağı için parameterlerin argüman olarak verilmesi pratik değildir.
- › Fakat genelde tek NAND cihazı vardır. Bundan dolayı NAND parametreleri argüman olarak verilir.

80

Gömülü Linux eğitimleri, kuruluđu ve desteđi

<http://ucanlinux.com>

*ubifs*

- NAND üzerinde çalışırken mutlaka 4 parametrenin el altında olması gerekir.

```
PEB : 16KiB -p
LEB : 15872 -e
Min I/O Size : 512 -m
Sub page size: 256 -s
```

- Bu parametrelerinin bazıları ubifs bazıları ubi için gereklidir.
- Örneğin, ubifs PEB ile ilgilenmez, benzer şekilde ubi sistemi de LEB ile ilgilenmez.
- ubifs, ubi üzerine kuruludur. ubi sistemi bir dosya sistemi değildir, dosya sistemi ile bare NAND arasında bir katmandır.

81

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ubifs*

- ubi'nin esas amacı bad blokları yönetmektir. Böylece ubi üzerine kurulu dosya sistemleri bu sıkıntılı işlerle uğraşmazlar.
- ubi üzerine illa ki ubifs kurmak gerekli değildir. squashfs, cramfs gibi dosya sistemleri ubi üzerine kurulabilir.
- Örnek bir ubifs kuruluşu örnekte verilmiştir. ubifs ile NAND sisteminin hiç bir ilgisi yoktur. Bu ilgi ubinize komutu ile kurulur.

```
$ mkfs.ubifs -v -m 512 -e 15872 -r RootFS -c 4228 root.ubifs
```

```
ubifs parameterleri:
LEB : 15872 -e
Min I/O Size : 512 -m
```

82

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*ubifs*

```
$ cat root.ini

[root] # Keyfi bir isim.
mode=ubi # Zorunlu bir satır.
image=root.ubifs # mkfs.ubifs ile elde edildi.
vol_id=0 # Kaçınıcı ubi volume?
vol_size=32MiB # Kaba bir boy miktarı.
vol_type=dynamic # LEB'ler rw olabilir. ro=static
vol_name=root # Volume name.
vol_flags=autoresize # İlk mount'da genişle.

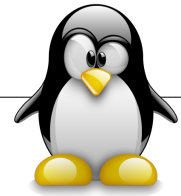
$ ubinize -v -o root.ubi -p 16KiB -m 512 -s 256 root.ini

ubi paramterleri:
PEB : 16KiB -p
Min I/O Size : 512 -m
Sub page size: 256 -s

root.ini dosyası nandwrite, nanddump, ubiformat ile mtd'ye
yazılabilir.
```

83

Gömülü Linux eğitimleri, kuruluşu ve desteği

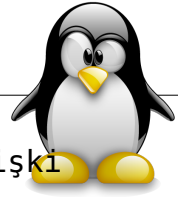
<http://ucanlinux.com>*ubifs*

- > Bir ubi sisteminde en fazla 128 volume, diğer bir deyişle 128 adet ubifs olabilir.
- > Gömülü sistemlerde genelde her ubi için bir adet ubifs bulunur.  
Birden fazla olması pek nadirdir.
- > Bir mtd cihazı ubiformat ile formatlanmamışsa, ubiattach her zaman başarısız olur.
- > ubiattach, her zaman bare NAND cihazı ile ubi cihazını birbirine bağlar.  
Kaba bir deyişle, PEB ve LEB arasındaki ilişkiyi sağlar.
- > LEB boyları, her zaman PEB'den biraz daha kısadır.
- > mtd ile ubi arasındaki bağlantı ubidetach -m No komutu ile koparılır.

84

Gömülü Linux eğitimleri, kuruluşu ve desteği

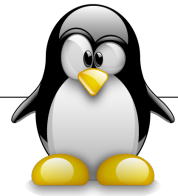
<http://ucanlinux.com>

*ubifs*

- > umount edilmeden, yani ubi ile ubifs arasındaki ilişki kesilmeden, ubidetach işlemi yapılamaz.
- > Her zaman önce ubidetach sonra umount yapılır.
- > NAND'a ne yazılırsa yazılsın, ilgili bölge MUTLAKA önce flash\_erase veya nanderase gibi komutlarla silinmelidir.
- > Silme işlemi bütün bölüme 1 yazar.
- > Yazma işlemi asla 0'dan 1'e yazım yapamaz. Bundan dolayı önce erase işlemi ile bütün bölüm 1 yapılır.
- > write işlemi sadece 1'den 0'a değişimleri yapabilir.
- > NAND'a yazılacak her dosya mutlaka minimum I/O değerinin bir katı olmalıdır.

85

Gömülü Linux eğitimleri, kuruluşu ve desteği

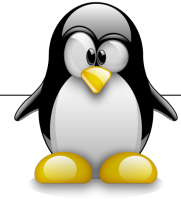
<http://ucanlinux.com>*ubifs*

- > Bir dosya boyunu 2k'nın katı yapmak için,  
\$ dd if=foo of=foo2k bs=2k conv=sync
- > Bir volume ilk defa mount ediliyorsa ve içi tamamen boşsa, ubifs sistemi bunu anlar ve otomatik olarak ubifs dosya sistemini yaratır.  
  
\$ mount -t ubifs ubi3:root /mnt/root  
UBIFS: default file-system created <-- Sadece ilk mount işleminde uygulanır.  
...
- > mtd\_debug programı ile bare NAND hakkında bilgi alınabilir.  
  
\$ mtd\_debug info /dev/mtd3

86

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*ubifs*

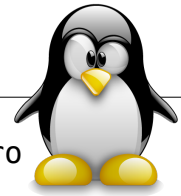
- › Diğer faydalı komutlar ve dosyalar:

```
$ mtdinfo --all
$ ubiinfo /dev/ubi0
$ ubiinfo /dev/ubi0_0
$ cat /proc/mtd
$ cat /proc/partitions
$ cat /proc/cmdline
$ ls /proc/fs/ubifs
$ ls /proc/drivers/mtd
$ ls /proc/drivers/mtd/ubi
```

- › PEB'lerin %1'i bad block yönetimi için kullanılır.
- › Bir PEB hatalı olursa hemen yeni bir PEB atanır. Hatalı olan bad diye işaretlenir. LEB, yeni PEB'e göre güncellenir. Bad block yönetimini ubi yapar, ubifs değil.

87

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ubifs*

- › Eğer vol\_type= static ise mount işlemi her zaman ro yapılır. vol\_type= dynamic ise mount işlemi ro veya rw yapılabilir.
- › Eğer rw mount işleminde düzeltilemez bir hata varsa, mount işlemi otomatik olarak ro yapılır.
- › /dev/mtd0 veya bare NAND flash cihazlarına yazım için dd kullanılamaz. Çünkü dd, bad block'ları kontrol etmez. Fakat okuma yapılabilir.
- › Yorum /dev/ubi0\_1 için:  
/dev/ubi0'ı /dev/sda gibi düşün.  
/dev/ubi0\_1'i /dev/sda1 gibi düşün.

88

Gömülü Linux eğitimleri, kuruluşu ve desteği

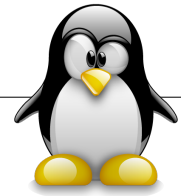
<http://ucanlinux.com>

*ubifs*

- mtd üzerinde ubi varsa asla mtd üzerinden okuma/yazma yapılmamalıdır.  
Çünkü ubi'nin kendi iç veri yapısı vardır.  
Yazma işlemi bu yapıyı bozar.  
Okuma işlemi ile bu yapı okunabilir ama anlamsızdır.
- /dev/mtd0 her zaman bare flash'a denk gelir.  
Bare flash, host tarafında ubinize tarafından hazırlanır.
- Tamamen boş dosya sisteminde 0xFF vardır.  
\$ dd if=/dev/ubi0\_0 of=foo
- ubi imajlarının loop mount özelliği yoktur.  
Fakat aynı etki nand simülatörü kullanılarak elde edilebilir.

89

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ubifs*

- Host sisteminde ubi imajının incelenmesi
  - 128MB'lik disk kur /tmp/mtd7.img imajını mount et.
- ```
$ modprobe nandsim first_id_byte=0xec second_id_byte=0xa1
third_id_byte=0x00 fourth_id_byte=0x15

$ dd if=/tmp/mtd7.img of=/dev/mtd0 bs=2048

$ modprobe ubi mtd=0

$ mount -t ubifs /dev/ubi0_0 /mnt/ubifs

$ df /mnt/ubifs
$ ls -l /mnt/ubifs

$ umount /mnt/ubifs
$ rmdir ubifs ubi nandsim
```

90

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

ubifs

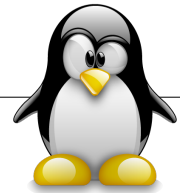
Kernel Support:

```
Device Drivers --->
<*> Memory Technology Device (MTD) support --->
    [*] Command line partition table parsing

<*> NAND Device Support --->
    <*> NAND Flash device on OMAP2, OMAP3 and OMAP4
```

91

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*nandsim*

```
$ modinfo nandsim
```

NANDsim parametreleri:

```
modprobe nandsim first_id_byte=0x20 second_id_byte=0x33 (16MiB, 512 bytes page)
modprobe nandsim first_id_byte=0x20 second_id_byte=0x35 (32MiB, 512 bytes page)
modprobe nandsim first_id_byte=0x20 second_id_byte=0x36 (64MiB, 512 bytes page)
modprobe nandsim first_id_byte=0x20 second_id_byte=0x78 (128MiB, 512 bytes page)
modprobe nandsim first_id_byte=0x20 second_id_byte=0x71 (256MiB, 512 bytes page)

modprobe nandsim first_id_byte=0x20 second_id_byte=0xa2 third_id_byte=0x00
                    fourth_id_byte=0x15 (64MiB, 2048 bytes page)

modprobe nandsim first_id_byte=0xec second_id_byte=0xa1 third_id_byte=0x00
                    fourth_id_byte=0x15 (128MiB, 2048 bytes page)

modprobe nandsim first_id_byte=0x20 second_id_byte=0xaa third_id_byte=0x00
                    fourth_id_byte=0x15 (256MiB, 2048 bytes page)

modprobe nandsim first_id_byte=0x20 second_id_byte=0xac third_id_byte=0x00
                    fourth_id_byte=0x15 (512MiB, 2048 bytes page)

modprobe nandsim first_id_byte=0xec second_id_byte=0xd3 third_id_byte=0x51
                    fourth_id_byte=0x95 (1GiB, 2048 bytes page)
```

92

Gömülü Linux eğitimleri, kuruluşu ve desteği

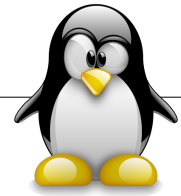
<http://ucanlinux.com>

pseudo file systems

- › Pseudo dosya sistemleri kernel ve user space arasında iki yönlü bilgi alışverişi için tasarlanmıştır.
- › Her zaman ya boot anında ya da tam kullanım anında kurulurlar ve sistem kapanınca yok olurlar.
- › Fiziksel olarak bir diskte kurulmazlar bundan dolayı dosya boyları her zaman 0'dır ama içleri dolu olabilir.
- › Gömülü sistemlerde genelde açılış betikleri tarafından mount edilirler.
- › Çok kullanılan bazı dosya sistemlerinin mount edilmesi...

93

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*pseudo file systems*

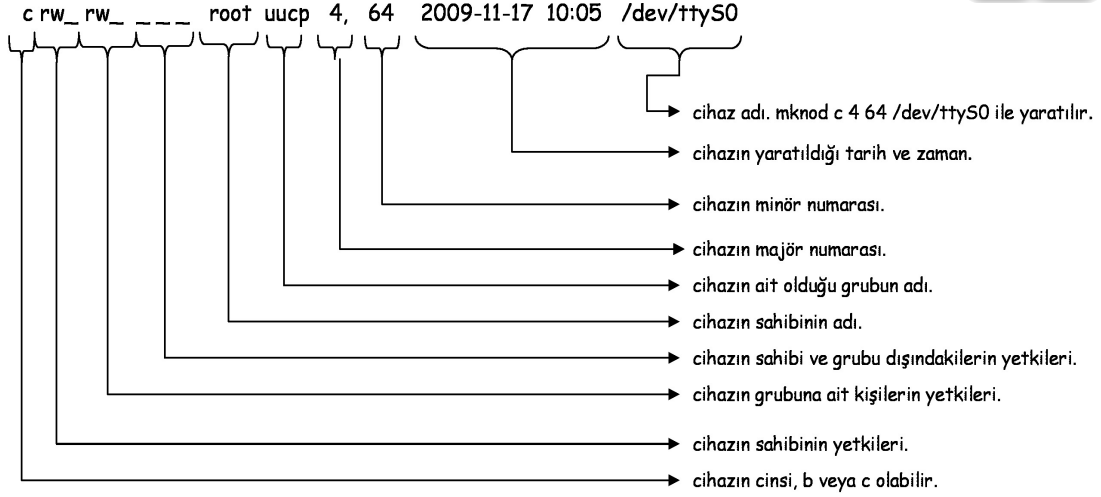
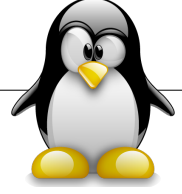
```
mount -t proc   proc   /proc
mount -t sysfs sysfs  /sys
mount -t devpts devpts /dev/pts
mount -t tmpfs tmpfs  /dev/shm
```

```
echo /sbin/mdev > /proc/sys/kernel/hotplug
mdev -s
```

94

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

VFS, /dev directory

95

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*VFS, switch table*

Character Device Functions Switch Table

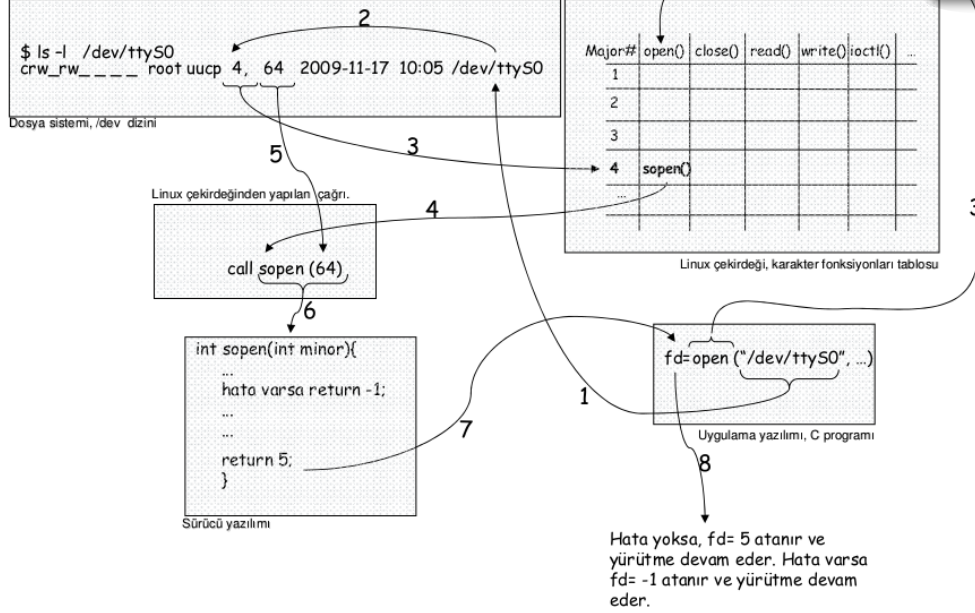
	Major#	open()	close()	read()	write()	ioctl()	...
bellek aygıtları	1						
pseudo tty slave aygıtları	2						
pseudo tty aygıtları	3						
tty aygıtları	4	sopen()	sclose()	sread()	swrite()	siioctl()	
diğer aygıtlar	...						

96

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

VFS, open

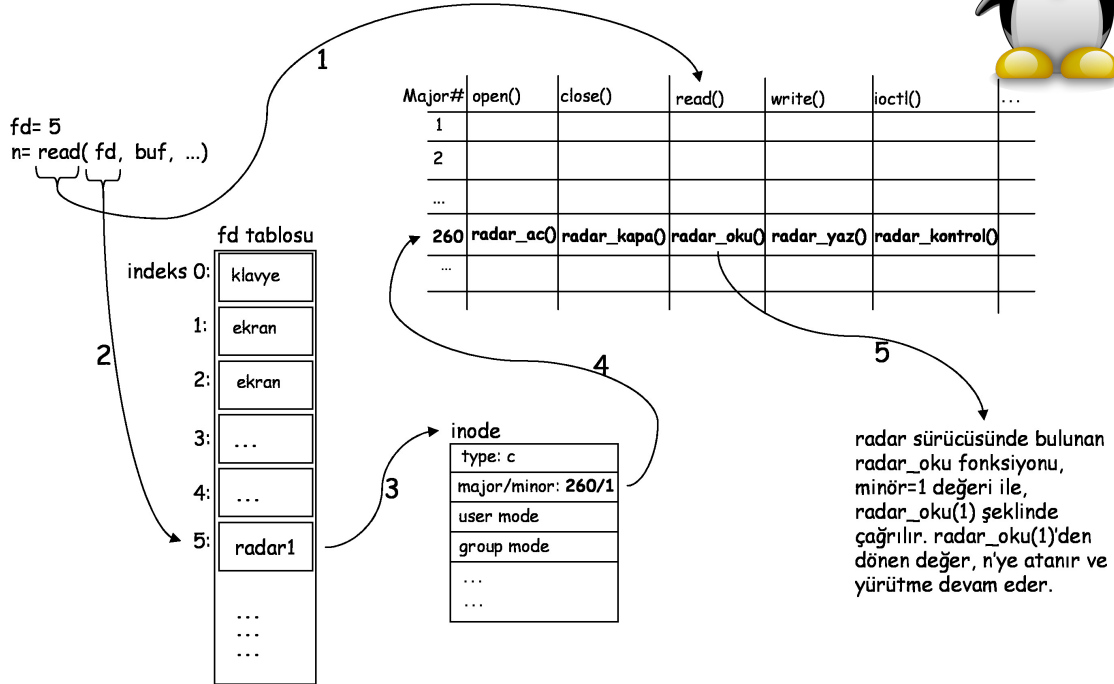
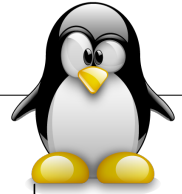


97

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

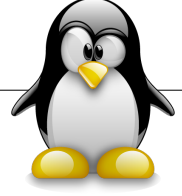
VFS, read



98

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



Bölümlendirmeler

99

Gömülü Linux eğitimleri, kurulumu ve desteği

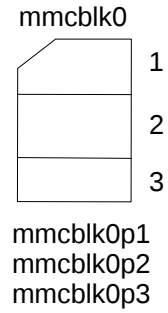
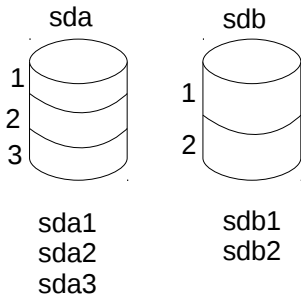
<http://ucanlinux.com>

Disklerin İsimlendirilmesi



SCSI disklerinin isimlendirilmesi

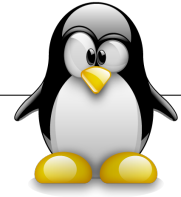
MMC/SD kartların isimlendirilmesi



100

Gömülü Linux eğitimleri, kurulumu ve desteği

<http://ucanlinux.com>

MS-DOS Partitions

- › Disklerin tek bir blok olarak kullanılması pratik değildir.
- › Diskler, daha ufak disklerle ayrılarak kullanılırlar. Bu ayırma işlemine bölümlendirme denir. Linux pek çok disk bölümlendirmesini destekler.
- › MMC diskler için MSDOS bölümlendirmesi, NAND diskler için mtdparts (NAND Disk Partition Types) bölümlendirilmesi kullanılır.
- › Her iki bölümlendirme desteği çekirdeğe verilmelidir.
- › MMC bölümlendirilmesi fdisk/sfdisk/cfdisk vs ile yapılabilir.

101

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*MS-DOS Partitions*

```
$ fdisk /dev/mmcblk0
```

veya loop cihazı olarak...

```
$ dd if=/dev/zero of=/tmp/mmc.img bs=1M count=32
```

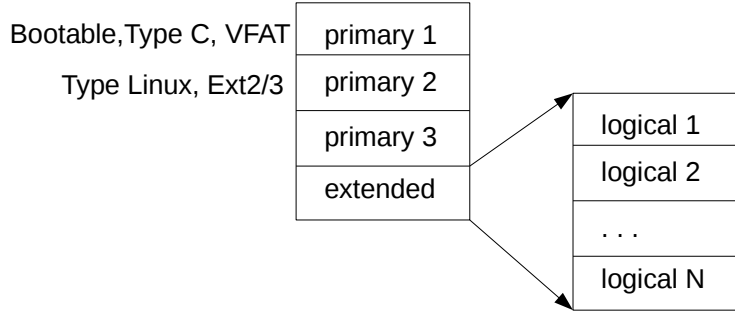
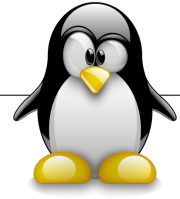
```
$ fdisk /tmp/mmc.img
```

- › Bütün işler bellekte yapılır, w ile çıkılmadığı sürece değişiklikler kaydedilmez.
- › x ile uzman moduna geçilir. Uzman modunda, başlangıç sektörü ayarlanabilir.
- › U-Boot için birinci sektör c tipinde ve bootable olmalıdır.
- › Gömülü sistemlerde, MMC üzerinde genelde 2 bölüm kullanılır. İlk bölüm u-boot'un desteklediği bir dosya sistemi olur, vfat/ext2/ext3/ext4 gibi.

102

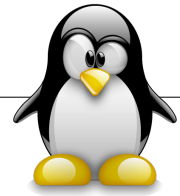
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

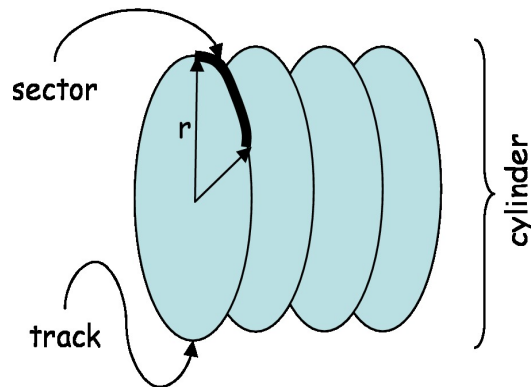
MS-DOS Partitions

103

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*MS-DOS Partitions*

```
$ fdisk [-b sectorsize] [-C cyls] [-H heads] [-S sects] device
```



104

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

MS-DOS Partitions

```
dd if=/dev/zero of=$MMC_DEVICE bs=512 count=1

sudo fdisk -H255 -S63 $MMC_DEVICE << EOF
o
n
p
1

+$BOOT_SIZE
a
l
t
c
x
b
l
63
r
n
p
2

+$ROOT_SIZE
p
w
EOF
```

105

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*MS-DOS Partitions*

Kernel Support:

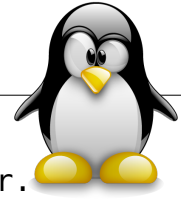
```
[*] Enable the block layer --->
    Partition Types --->
        [*] Advanced partition selection
        [*] PC BIOS (MSDOS partition tables) support
```

106

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

MTD Partitions



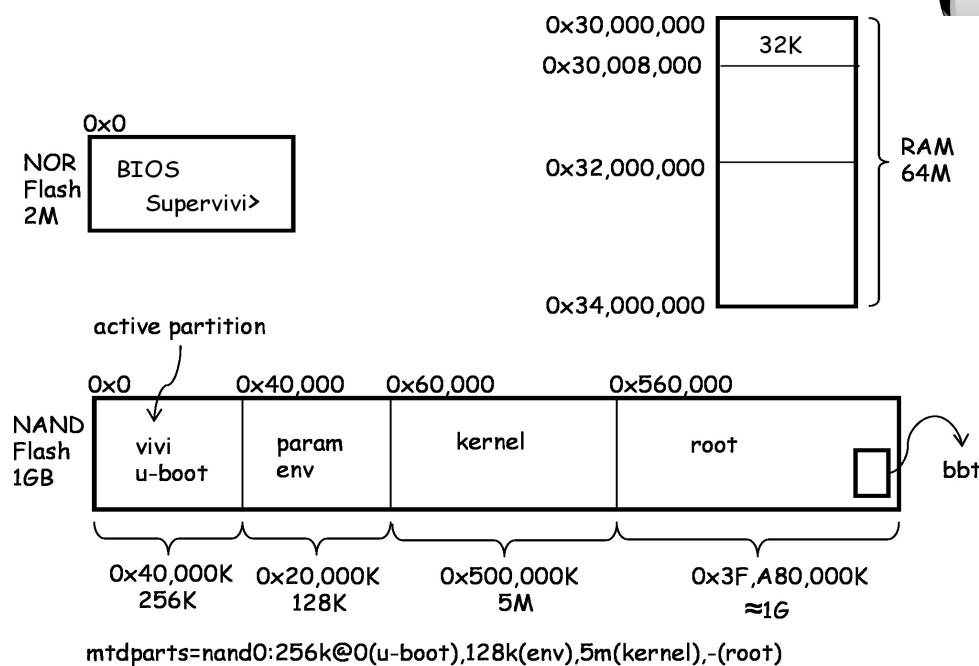
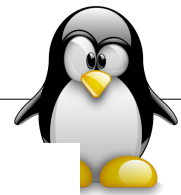
- > MS-DOS bölümlendirmesinde, bölümlendirme tablosu, diskin 1. sektöründe saklanır. Bu sektöre MBR denir. Çekirdek, MBR'yi okuyarak bölümlendirme hakkında bilgi sahibi olur.
- > Fakat NAND disklerde MBR kavramı yoktur. Çekirdek, mtdparts parametresi sayesinde NAND bölümlendirmesinden haberdar olur.
mtdparts değişkeni bir kaç yoldan çekirdeğe verilebilir.
- > En yaygın olanı u-boot çevre değişkeni olarak, bootargs içinde tanıtmaktır. (test sistemlerinde tavsiye edilir.)
- > Diğer yol, çekirdek derlemesi sırasında, boot parametresi olarak vermektir. (nihai sistemler için tavsiye edilir.)
- > Son yol ise, doğrudan u-boot kodu içine kazıdır. (nihai sistemler için tavsiye edilir.)

107

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

MTD Partitions



108

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

MTD Partitions



Kernel Support:

Device Drivers --->

```
<*> Memory Technology Device (MTD) support --->
[*] Command line partition table parsing
```

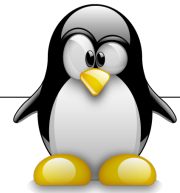
The format for the command line is as follows:

```
mtddparts=<mtdddef>[;<mtdddef>]
<mtdddef> := <mtdd-id>:<partdef>[,<partdef>]
<partdef> := <size>[@offset][<name>][ro]
<mtdd-id> := unique id used in mapping driver/device
<size> := standard linux memsize OR "-" to denote all
remaining space
<name> := (NAME)
```

109

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



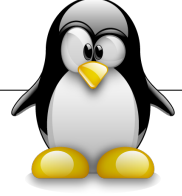
Özel Cihazlar

110

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Özel Cihazlar: /dev/null



- › Linux sistemini, sosuz kapasiteli çöp tenekesidir.

```
$ nohup prog 1>/dev/null 2>/tmp/prog.err &
```

```
$ dd if=/dev/sda of=/dev/null
```

111

Gömülü Linux eğitimleri, kuruluşu ve desteği

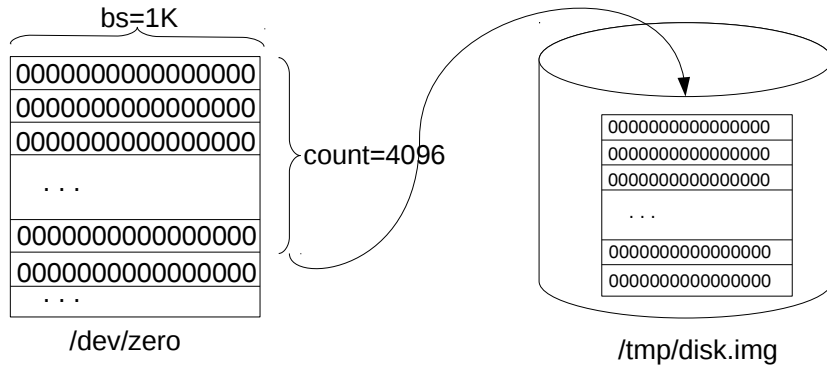
<http://ucanlinux.com>

Özel Cihazlar: /dev/zero



- › İçinde sonsuz adet “sıfır” bulunan bir dosyadır.

```
$ dd if=/dev/zero of=/tmp/disk.img bs=1K count=4096
```

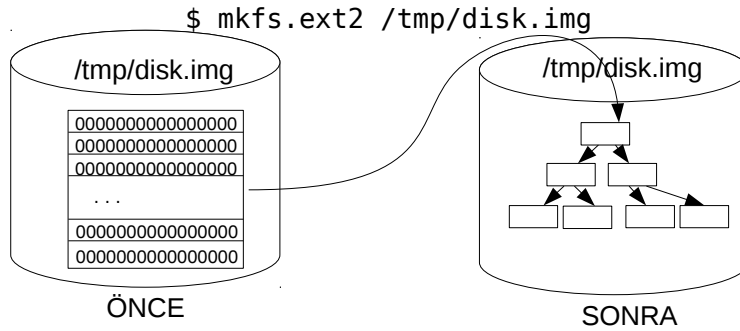


112

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Özel Cihazlar: /dev/zero



```

$ mkfs.ext2 /tmp/disk.img
$ mkdir /mnt/sanal.disk
$ mount -o loop /tmp/disk.img /mnt/sanal.disk
...
$ umount /mnt/sanal.disk
$ gzip disk.img

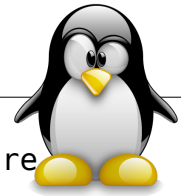
```

113

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Özel Cihazlar: /dev/random /dev/urandom



- /dev/random cihazı sistemdeki entropi değişimine göre keyfi sayı üretir.
- Sistemde entropi değişikliği yoksa sayı üretilmez. Bundan dolayı üzerinde iş yapılmayan makinelerde sayıların üretilme hızı çok yavaştır. Fakat nitelikli keyfi sayılar üretilir.
- /dev/urandom cihazı hızlı keyfi sayı üretir. Üretilen sayılar, keyiflik bakımdan çok da nitelikli değildir.

```

$ dd if=/dev/random
$ dd if=/dev/urandom of=/dev/sdc # DENEMEYİN !!!

```

114

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Özel Cihazlar: /dev/full



- Bu cihaz tam dolu bir cihazı temsi eder. Uygulama programlarında "tam dolu dosya" testi için kullanılabilir.
- Bu cihazdan veri okunabilir ama yazılamaz. Veri yazılacağı zaman cihaz dolu veya dosya dolu hatası alınır.

```
$ echo 123 > /dev/full  
$ echo $?
```

115

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

rsync

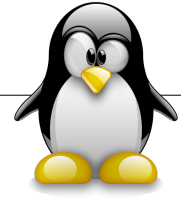


- Genelde farklı 2 makine içindeki dizinleri eş tutabilmek için kullanılır.
- Özellikle gömülü sistemlere program aktarmak veya güncelleme yapmak için çok elverişlidir.
- Çok düşük band genişliği kullanılır. Sadece güncellemeleri taşır. Binary dosyaların dahi sadece farklılıklarını taşır.
- İki dosya aynı ise checksum denetim ile karar verir.
- Taşıma sırasında sıkıştırma yapar. Bütün dosya özelliklerini korur.
- En önemlisi ssh üzerinden işler, güvenlidir.

116

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

rsync

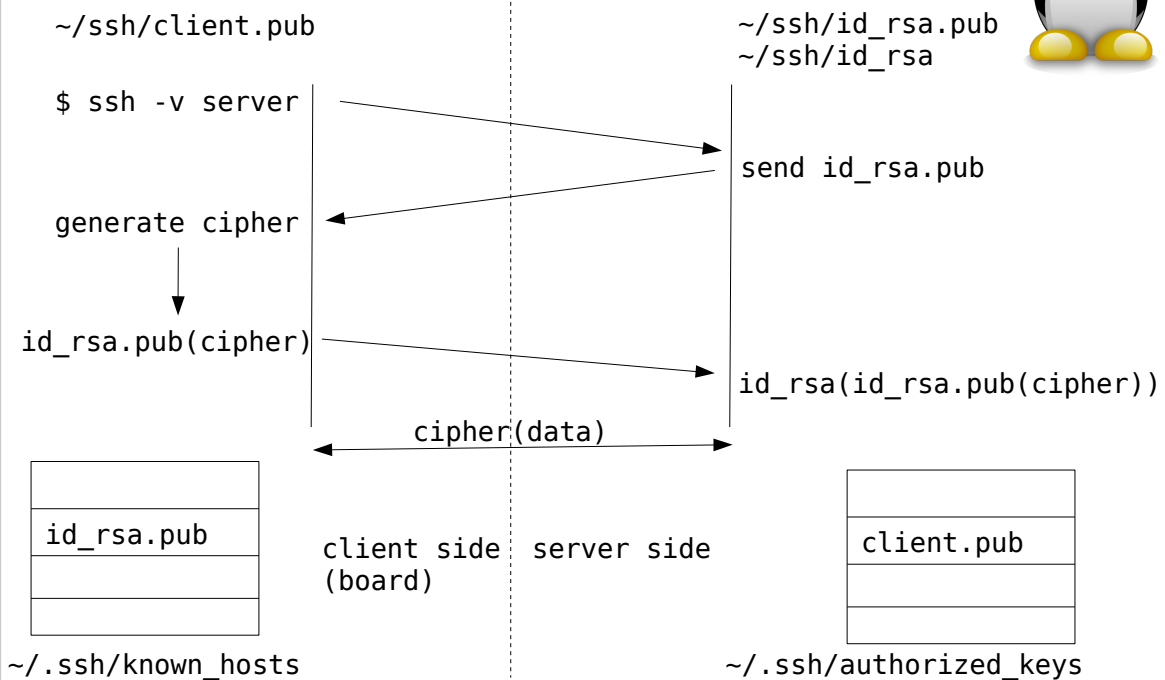
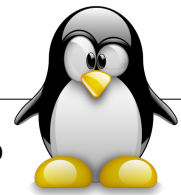
```
# -a: archive mode.
# -v: verbose on
```

```
$ rsync arch/arm/boot/uImage root@10.1.2.3:/boot
$ rsync -a RootFS/etc 10.1.2.3:/
```

- Uzak makinelere kopyalama sırasında, uzak makinedeki kullanıcının şifresi sorulacaktır.
- Uzaktaki makineye public anahtar kaydedilirse şifre sorulmaz.

117

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ssh*

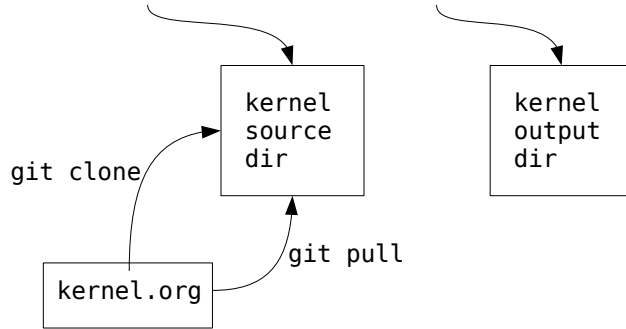
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

makefile

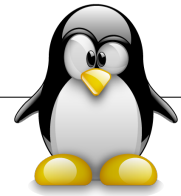
u-boot, kernel, busybox ve buildroot derlenirken
-C ve O= seçeneklerinin kullanılması önemle tavsiye edilir.

```
$ .../make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabihf
-C source_code_dir O=compiled_output_dir
```



119

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*git*

- › Linux topluluğu tarafından kullanılan en yaygın proje dağıtım ve sürüm kontrol sistemidir.
- › Temel olarak ssh sistemini kullanır.
Artık neredeyse bütün yazılımlar git üzerinden dağıtılmaktadır.
Şimdilik clone/pull kullanımını bize yeterli olacaktır.

```
$ git clone git://git.buildroot.net/buildroot
$ git pull
$ git branch
$ git branch -r
$ git branch -a
$ git checkout -b builroot-at91
```

- clone edilen dizin geliştirme için kullanılmamalıdır.
-C ve O= ile derleme yapılmalıdır.

120

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

git

- clone edilen dizin geliştirme için kullanılmamalıdır. -C ve O= ile derleme yapılmalıdır.
- buildroot paketinin, /project/buildroot altına clone ile alındığını kabul edelim. buildroot sistemi aşağıdaki gibi derlenebilir ve sonuçlar /nk/workspace/out içine atılır. Böylece orijinal buildroot sisteminde hiç bir güncelleme yapılmaz. Daha sonra git pull ile sorunsuzca güncelleme yapılabilir.
- Ayrıca birden fazla proje aynı buildroot sisteminin kaynak kodunu birbirlerine karıştırmadan kullanabilir.

```
$ make -C /project/buildroot O=/nk/workspace/out menuconfig
$ make -C /project/buildroot O=/nk/workspace/out
```

121

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*/etc/inittab*

- Çekirdek, kök dosya sistemini bağladıktan sonra, genelde /sbin/init programını işletir. Bu program sistemi ayağa kaldırır. Sistemin ayağa kalkma şekli /etc/inittab dosyası tarafından tespit edilir.
- busybox'ın kullandığı inittab dosyası standard inittab dosyasından çok farklıdır.
- Run Level kavramı yoktur. Zaten gömülü sistemler için bu kavrama gerek de yoktur.
- CM-T3517 ARM makinesinden bir örnek sonraki sayfada verilmiştir.

122

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

/etc/inittab

```
nazim@nkoc:/tftpboot/RootFS/etc$ cat inittab

# Startup the system
null::sysinit:/bin/mount -t proc proc /proc
#null::sysinit:/bin/mount -o remount,rw / # REMOUNT_ROOTFS_RW
null::sysinit:/bin/mkdir -p /dev/pts
null::sysinit:/bin/mkdir -p /dev/shm
null::sysinit:/bin/mount -a
null::sysinit:/bin/hostname -F /etc/hostname

# now run any rc scripts
::sysinit:/etc/init.d/rcS

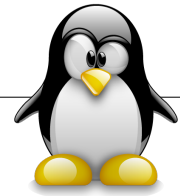
# Put a getty on the serial port
tty02::respawn:/sbin/getty -L tty02 115200 vt100 # GENERIC_SERIAL

# Stuff to do for the 3-finger salute
::ctrlaltdel:/sbin/reboot

# Stuff to do before rebooting
null::shutdown:/etc/init.d/rck
null::shutdown:/bin/umount -a -r
null::shutdown:/sbin/swapoff -a # !!!
```

123

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*/etc/passwd*

```
$ ls -l passwd
-rw-r--r-- 1 root root 596 May 31 23:29 passwd
```

```
$ ls -l shadow
-rw----- 1 root root 380 Ağu  4 09:38 shadow
```

Örnek satırlar:

```
root:x:0:0:root:/root:/bin/sh
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
```

```
default:x:1000:1000:Default non-root user:/home/default:/bin/sh
```

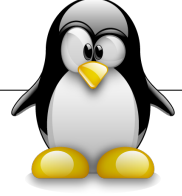
```

↓           ↓           ↓           ↓           ↓           ↓           ↓
login name  encrypted password  user ID  group ID  comment field  home directory  user command
```

124

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

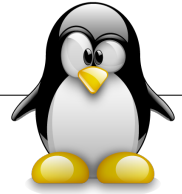
/etc/passwd

- > Şifre kolonu boşsa, kullanıcı şifresiz girebilir.
- > Şifre kolonunda x varsa, şifre shadow içinde saklıdır.
- > Son kolon çok önemlidir. Login/Passwd geçildikten sonra çalıştırılacak komuttur. Genelde /bin/sh veya /bin/false seçilir. Buradaki değer, SHELL değişkenine otomatik olarak atanır. Boşsa /bin/sh kabul edilir.
- > HOME değişkeni, "home directory" kolonu olarak atanır. Giriş yapıldıktan sonra geçilecek ilk dizindir.

```
$ man 5 passwd
$ man passwd
$ man adduser
```

125

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*/etc/shadow*

- Şifre bigisi ve sürelerini saklar. Sadece root erişebilir. passwd dosyasındaki her bir şifreli kullanıcı için burada bir kayıt olmalıdır.

```
$ cat shadow
```

```
root:$1$V4H6aNhe$jZWUzp9i08sJVIXpYBjEH.:10933:0:99999:7:::
bin:!:10933:0:99999:7:::
daemon:!:10933:0:99999:7:::
adm:!:10933:0:99999:7:::
lp:!:10933:0:99999:7:::
sync:!:10933:0:99999:7:::
shutdown:!:10933:0:99999:7:::
halt:!:10933:0:99999:7:::
uucp:!:10933:0:99999:7:::
operator:!:10933:0:99999:7:::
ftp:!:10933:0:99999:7:::
nobody:!:10933:0:99999:7:::
default:!:10933:0:99999:7:::
```

126

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

/etc/shadow

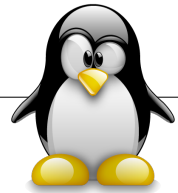
```
root:$1$V4H6aNhe$jZwUzp9i08sJVIXpYBjEH.:10933:0:99999:7:::
```

login şifre
name ! veya * varsa passwd ile giriş zaman bilgileri yapılamaz. Ama başka türlü login olabilir. Şifresiz giriş için boş bırakılabilir. ! ile başlarsa şifre kilitlidir.

```
$ man 5 shadow
$ man passwd
```

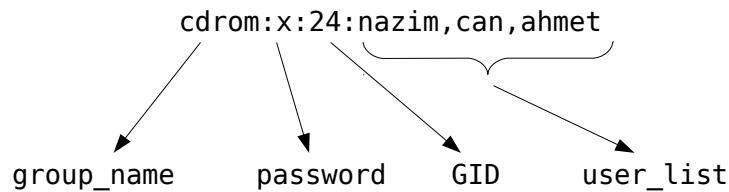
127

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*/etc/group*

```
$ cat /etc/group
```

```
adm:x:4:nazim
tty:x:5:
disk:x:6:
cdrom:x:24:nazim,can,ahmet
```

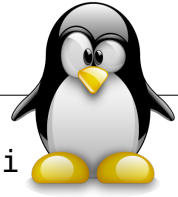


```
$ man 5 group
$ man groupadd
```

128

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

/etc/init.d/

- > buildroot sistemi aşağıdakine benzer bir rcS betiği ile sistemi ayağa kaldırır.
- > rcK betiği ile sistemi kademeli olarak, açılışın ters sırasına göre kapatır.

```
$ cat /etc/init.d/rcS
```

```
#!/bin/sh
```

```
for i in /etc/init.d/S??*
do
    $i start
done
```

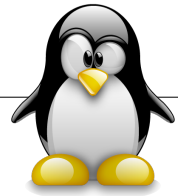
```
$ cat /etc/init.d/rcK
```

```
#!/bin/sh
```

```
for i in $(ls -r /etc/init.d/S??*)
do
    $i stop
done
```

129

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*/etc/init.d/*

```
#!/bin/sh
```

```
case "$1" in
    start)
        echo "Starting network..."
        /sbin/ifup -a
        ;;
    stop)
        echo -n "Stopping network..."
        /sbin/ifdown -a
        ;;
    restart|reload)
        "$0" stop
        "$0" start
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
esac
exit $?
```

130

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Atomic Update



- Prensipte olarak bütün dosya sistemleri RO bağlanmalıdır.
- RW olması gereken dosyalar tmpfs'e yönlendirilebilir. Fakat güç kesilince bu dosyalar yok olur.
- Uygulamaların config dosyaları gibi bazı dosyaların kalıcı olması istenir.
- Kalıcı dosyalar illa ki RW modunda bir bağlı bir bölüm üzerinde tutulmalıdır.
- RW modunda bağlı bir bölüm üzerinde güvenli güncelleme yapabilmek için "atomic update" denilen çok basit bir yöntem uygulanabilir.
- Bu yöntem ani kapanma gibi durumlarda dosyanın bütünlüğünü korur.

131

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Atomic Update



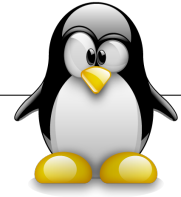
- Unutulmamalıdır ki ubifs gibi kapanmaya dayanıklı sistemler dosya değil file system bütünlüğünü sağlarlar.
- Diğer bir deyişle, ani kapanmalardan sonraki mount işleminin kesin olarak yapılabileceğini garantiler ama dosya içeriği konusunda bir garanti sunmazlar.
- Dosya içeriğini korunması konusundaki garanti "atomic update" ile sağlanır.
- Ubifs ile dosya sistemi ve "atomic update" ile de dosya bütünlüğü korunmuş olur.
- Canlı dosyalar veya büyük dosyalar bu tekniğe uygun değildir.

132

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Atomic Update



- > F1 dosyası güncellenecek olsun.

```
F2= copy(F1)
update(F2)
sync
mv F2 F1
```

- > Bu yöntemde, ani kapanmalarda F1 bozulmaz. Çünkü bütün güncelleme F2 üzerinde yapılır.
- > Sadece, 4. adımda dosya ismi değiştirme sırasında tehlike vardır. Fakat posix standardı, rename veya move işleminin atomik olması gerektiğini söyler ve ani kapanmalarda mv işlemi F1'i bozamaz.
- > ubifs ve diğer pek çok dosya sistemi posix standardını sağlar.

133

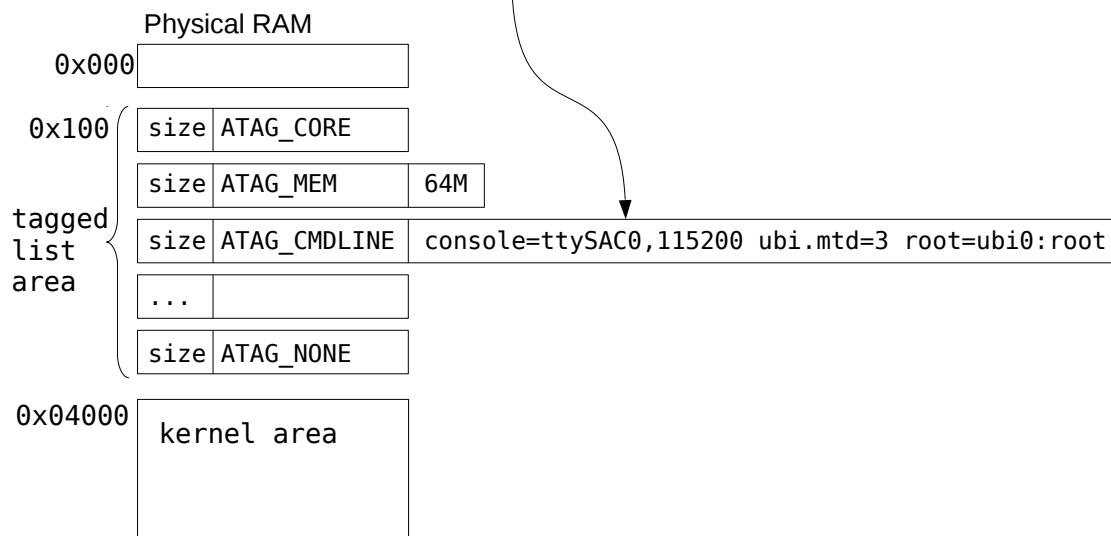
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kernel Parameters Area



```
U-Boot
bootargs="console=ttySAC0,115200 ubi.mtd=3 root=ubi0:root"
```

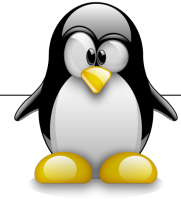


134

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kernel Parameters Area



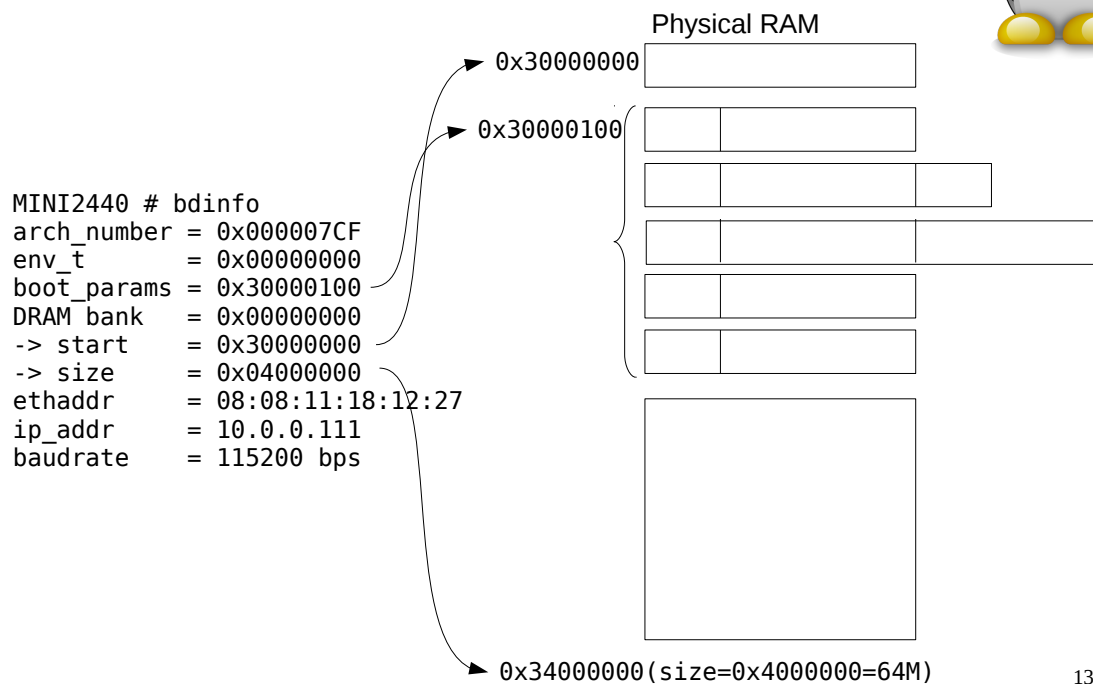
- Parametre listesi fiziksel RAM belleğin başlangıç adresinden sonra, tam 0x100 adresinden başlar.
- Parametre listesinin ilk elemanı CORE ve son elemanı NONE olmak zorundadır.
- MEM zorunludur.
Diğer etiketlerin hiç bir olmayabilir.
- Parametre listesi 0x4000 adresini geçmemelidir.
Linux gözü kapalı biçimde 0x4000 adresinden sonrasını kullanmaya başlar, parametreler ezilebilir.
- Parametreler u-boot tarafında bootargs ile, çekirdek derlemesinde "boot parameters" ile, modül yüklerken modül parametreleri şeklinde verilebilir.

135

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kernel Parameters Area



136

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Modül Parametreleri



- › insmod sırasında modüllere parametreler var=val şeklinde atanabilir.
- › Modül olmayan, derleme sırasında çekirdeğe doğrudan eklenen cihazlara ise parametreler modül_ismi.var=val şeklinde verilir.
- › Dinamik modüllere, henüz yüklü olmadığı için açılış sırasında parametre aktarılamaz.
- › `$ insmod usbcore blinkenlights=1`
- › `> setenv bootargs "... usbcore.blinkenlights=1"`
- › Yürütme zamanında değişiklik için `/sys/module/usbcore/parameters`

137

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Modül Parametreleri



Örnek:

```

$ lsmod

$ modinfo snd

> setenv bootargs " ... snd.debug=0"

$ modprobe snd debug=0

$ cd /sys/module/snd/parameters
$ ls -l

$ cat debug

$ echo "1" > debug # Diğer dosyalar ro durumundadır.

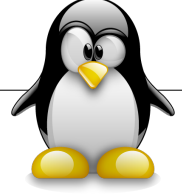
```

138

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



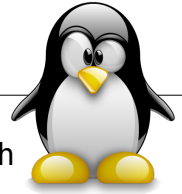
- Açılış teknikleri "çekirdek" ve "kök dosya sistemi"nin oturduğu yere göre çok fazla çeşitlilik gösterir.
- Her gömülü Linux projesinde, projeye uygun bir açılış tekniği tespit edilmelidir.
- Bunun istisnası test sistemleridir. Mümkünse test sistemleri her zaman ağ üzerinden kurulmalı ve test edilmelidir. Bu sayede geliştirme zamanı inanılmaz ölçüde kısalır.
- Gömülü sistemlerde çekirdek ve kök dosya sisteminin oturduğu cihazlar genelde "ağ", "NAND flash" ve "MMC" lerdir.

139

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



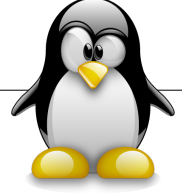
- Müşteriye verilecek sistemlerde mümkünse NAND flash üzerinde ubi+ubifs kullanılmalıdır.
- Mümkünse müşteriye verilecek sistemlerde MMC cihazı, asıl "Non Volatile Memory" olarak kullanılmamalıdır. MMC'lerin ömrü ve üzerindeki dosya sistemleri ubi sistemleri ile yarışamaz.
- Eğer gömülü sistem bir ağ ortamında ise, otellerdeki paralı film sistemleri gibi, sistemin tamamen ağ ortamından boot edip çalışması sağlanmalıdır. Böylece hem yönetim hem de gömülü sistemin maliyeti çok azalacaktır.
- Test sistemi için en ideal ortam ağ ortamı ise, müşteriye verilecek sistem için de en ideal ortam çekirdek ve ayrık derlenmiş initramfs sistemidir (ağ ortamında değilse)

140

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



1) Standard RootFS tekniği

- Bu yöntemde RootFS kurulur ve NAND veya MMC'de uygun bir bölüme kopyalanır.
- Ya da RootFS ağ üzerinden mount edilir.
- Çekirdek de ya ağ üzerinden çeşitli yöntemlerle alınır(rarp, bootp,dhcp,tftp gibi) alınır ya da NAND ve MMC üzerinden u-boot tarafından yüklenir.
- u-boot genelde NAND veya MMC üzerinde oturur.

RootFS \ kernel	Net	NAND	MMC
Net	net.net(test)		
NAND		nand.nand	
MMC			mmc.mmc

141

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



nand.nand örneği:
kernel(uImage) ve RootFS NAND içinde oturmaktadır.

NAND

u-boot	mtd0 (raw)
env	mtd1 (raw)
uImage	mtd2 (raw)
RootFS	mtd3 (ubi+ubifs)
...	

142

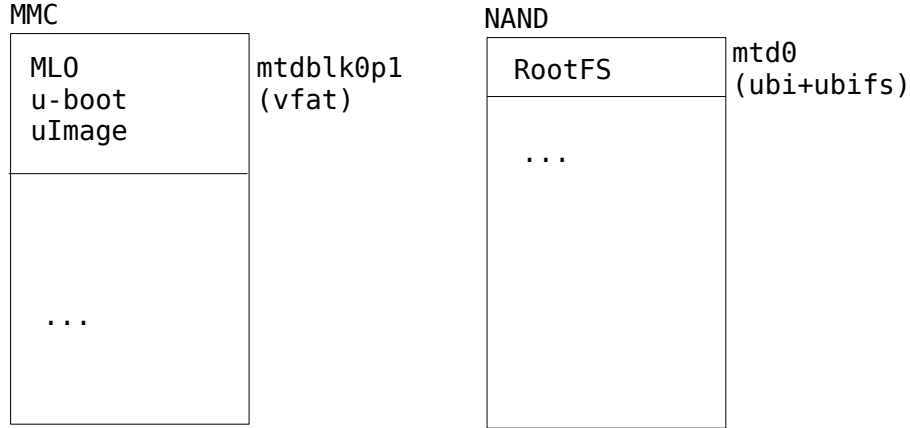
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



mmc.nand örneği: (zorlama bir örnek!)
kernel MMC'de, RootFS nand'da oturmuştur.



143

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



2) Intermedite RootFS (initramfs) tekniği.

- › Bu teknikte RootFS, doğrudan sanal bellekte (RAM+swap) kurulur.
- › Güç kesilince bütün bilgiler kaybolur.
- › Kalıcı olması gereken bilgiler MMC, NAND veya ağ ortamında saklanabilir.
- › cpio tekniği ile kurulur.
- › Büyük RootFS'ler için bu teknik uygun değildir.
- › Çekirdek açılır açılmaz, RootFS çok hızlı bağlanır.
- › RootFS, her zaman sıkıştırılmış tek bir dosyadır. Bu dosya çekirdeğe gömülür veya ayrı kullanılabılır.

144

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



2.a) Gömülü initramfs tekniği

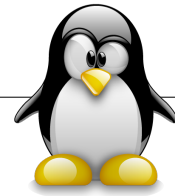
- Bu yöntemde RootFS dosyası çekirdeğin kodu içine, çekirdek derlemesi sırasında gömülür.
- Çekirdeğe gömülen her program GPL olmak zorundadır. Dolayısı ile RootFS içindeki her programın GPL olması gerekir.
- RootFS'deki her güncelleme için çekirdek yeniden derlenmelidir.
- uImage+RootFS tek bir dosya olacağı için, inanılmaz derecede hızlı açılan, basit bir sistemdir.
- Bütün sistemin yedeğini tutmak ve sistemin bütünü güncellemek çok kolaydır.
- Herhangi bir dosya sistemi desteğine gerek yoktur.

145

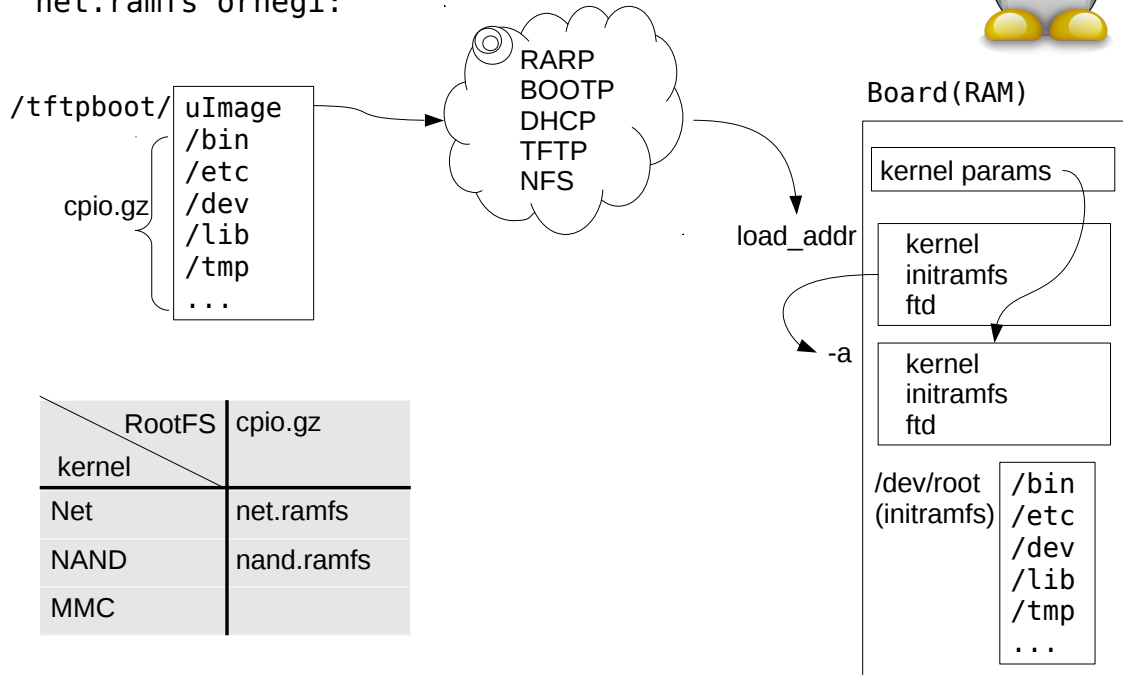
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



net.ramfs örneği:



Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



2.b) Ayrık initramfs tekniği

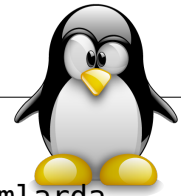
- › Bu yöntemde RootFS dosyası çekirdekten ayrı kurulur.
- › RootFS çekirdekten ayrı olduğu için, RootFS içindeki programlar özel lisansa sahip olabilir.
- › RootFS güncellendiğinde çekirdeğin derlenmesine gerek yoktur.
- › initramfs her zaman u-boot tarafından yüklenir. u-boot önce çekirdeği sonra initramfs'i yükler. İkinci yükleme fazladan vakit alır, ama açılış yine çok hızlıdır.
- › RootFS'i güncellemek artık daha kolaydır.
- › Herhangi bir dosya sistemi desteğine gerek yoktur.

147

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



- › cpio.gz dosyası genelde çekirdeğin bulunduğu ortamda bulunur. Fakat prensip olarak farklı ortamlarda da bulunabilir.
- › Örneğin çekirdek NAND'da bulunurken, cpio.gz dosyası ağda bulunabilir. Böyle bir durumda, u-boot tarafında, önce NAND'da bulunan çekirdek nboot ile addr1'e, ağda bulunan cpio.gz dosyası tftp ile addr2'ye yüklenip, "boot addr1 addr2" ile sistem açılabilir.

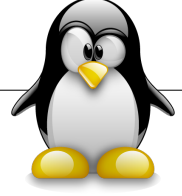
RootFS \	Net	NAND	MMC
kernel			
Net	net.cpio_net		
NAND		nand.cpio_nand	
MMC			mmc.cpio_mmc

148

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Açılış Teknikleri



nand.cpio_net örneği:

NAND

u-boot	mtd0 (raw)
env	mtd1 (raw)
uImage	mtd2 (raw)
...	

/tftpboot/cpio.gz



```
> nboot a uImage
> tftp b cpio.gz
> bootm a b
```

149

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot değişkenleri



- > autoload yes/no
no ise imaj yüklenmez, yes ise tftp ile imaj yüklenir.
Her durumda ağ yapılandırması yapılır.
Diğer bir deyişle tftp otomatik olarak işletilir.
Sadece rarpb, bootp ve hdcp için geçerlidir.
- > autostart yes/no
bootm otomatik olarak işletilir.
Sadece rarpb, bootp, dhcp, tftp disk, docb için geçerlidir.
- > baudrate
Konsolun hızı, varsayılan 115200
- > bootargs
cmdline bilgisi.
- > bootcmd
bootdelay sonrası çalışan autoboot betiği.

150

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot değişkenleri



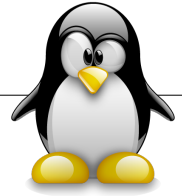
- > `bootdelay`
bootcmd'yi yürütmeden önceki bekleme süresi.
0, bekleme yok
-1, autoboot iptal edilir.
- > `bootfile`
tftp'nin yükleyeceği dosyanın ismi.
- > `ipaddr`
bord'un ip adresi, tftp kullanılır.
- > `loadaddr`
tftp'nin yükleme yapacağı bellek adresi.
- > `ethaddr`
Ethernet adresi, salt okunur.
- > `mtdparts`
mtd bölümlendirme tablosu, bootargs ile çekirdeğe gönderilir. u-boot ve kernel aynı tabloyu kullanır.

151

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot değişkenleri



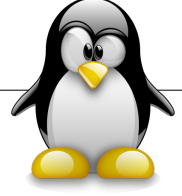
- > `serverip`
tftp sunucusunun bulunduğu adres.
- > Bazı env değişkenleri otomatik olarak bootp, dhcp ve tftp tarafından atanırlar.
Bu değişkenler genelde ağ yapılandırması ile ilgili olanlardır.
- > `filesize` değişkeni, en son kullanılan bootb, dhcp veya tftp ile komutları tarafından güncellenir.
- > Bunun dışında borda veya kullanıcıya özgü pek çok değişken olabilir.
- > Değişkenler 3 farklı biçimde saklanabilirler.

152

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot değişkenleri



1. u-boot değişkenleri NAND tarafında keyfi bir NAND bölümünde tutulabilirler..

Genelde bu bölüme env veya uboot-env isimleri verilir.

Bu bölüm her zaman raw formatındadır, dosya sistemi barındırmaz.

> set dynpart addr
komutu ile bu bölümün adresi u-boot'a tanıtılır.

> saveenv
komutu ile mevcut değişkenler env bölümüne kalıcı olarak yazılır.

> Eğer bu bölüm bozulursa, kaynak kodu içine gömülü olan değişkenler kullanılır.

153

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot değişkenleri



2. u-boot değişkenleri kaynak kodu içine gömülebilirler.

Eğer NAND/MMC'de bulunan değişkenler okunamaz ise, kod kod içinde gömülü olan değişkenler kullanılır ve bu durum açılışta ikaz mesajı olarak gösterilir.

Her borda ait header dosyası farklı bir yer ve isimde olabilir.

Kaynak kodunda, boards.cfg içinde desteklenen bordlar ve config dosyalarının isimleri bulunabilir.

env değişkenleri bbb için include/configs/am335x_evm.h dosyası içine yazılabilir. Bu dosya her bord için farklıdır.

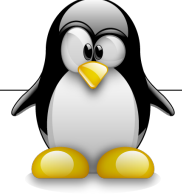
Nihai sistemde env kullanılmamalı, doğrudan header dosyası içinde tanım yapılmalıdır.

154

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot deęişkenleri



3. u-boot deęişkenleri MMC'den yüklenebilir.

Genelde, MMC 1. bölümünde, uEnv.txt dosyası içine yazılır. Açılış sırasında uygun bir betik uEnv.txt dosyası içindeki deęişkenleri okur.

uEnv.txt dosyası içinde ubootcmd komutu varsa, bu komutu otomatik işletir.

```
mmc 0:1 = /dev/mmcblk0p1
```

Yüklemenin genel yapısı aşağıdaki gibidir.

```
> load mmc 0:1 addr uEnv.txt  
> env import -t addr size
```

155

Gömülü Linux eğitimleri, kuruluşu ve desteęi

<http://ucanlinux.com>

net.net



İdeal Test Sistemi

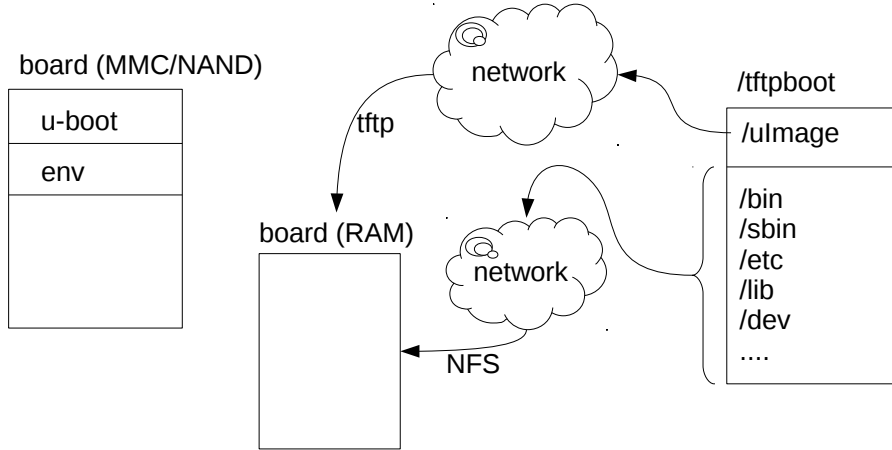
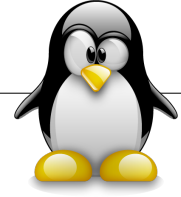
Çekirdek: Ağ üzerindedir, tftp ile yüklenir.
RootFS : NFS üzerinden mount edilir.

u-boot'un herhangi bir ortama yüklenmesi, NAND/MMC gibi yeterlidir.

156

Gömülü Linux eğitimleri, kuruluşu ve desteęi

<http://ucanlinux.com>

net.net

157

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*u-boot'un yüklenmesi**net.net*

- > Gömülü sistem için u-boot paketinin bir an için derlendiğini ve gerekli programların üretildiğini kabul edelim.
- > u-boot derlendikten sonra pek çok program üretilir.
- > MLO:
 - Farklı isimleri de olabilir.
 - u-boot.bin programının çok basit bir halidir.
 - Genelde u-boot.bin programını yükleyecek cihazları ayağa kalıdırır ve u-boot.bin programını yükler.
 - 1. seviye boot yükleyicisi de denir.
 - Genelde tek bir yükleyici her cihazdan boot edemez.
 - Seri, USB, Eternet, NAND, MMC'den boot edebilmek farklı farklı derlenir.
 - Her cihazda MLO olmayabilir.
 - MLO genel değildir ve borda bağlıdır.

158

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot'un yüklenmesi

net.net



- > u-boot.bin:
raw binary programdır.
ARM işlemcisi tarafından doğrudan yürütülür.
2. seviye yükleyicidir.
Cihazdan hemen hemen bağımsızdır, çok geneldir.
Borda ait bütün cihazları ayağa kaldırır.
Gömülü Linux projelerinde, u-boot'un ayağa kalkması çok önemli bir aşamadır.
Çünkü u-boot öncesi meydana gelen hataların tespit edilmesi çok zordur.
u-boot'tan sonrası kullanıcı ile çok iyi bir etkileşime sahiptir ve hatalar çok kolay tespit edilebilir.
- > u-boot.img
u-boot.bin programının u-boot imajı haline getirilmiş halidir.
u-boot.img= image_header + u-boot.bin
Bazı bordlar illa ki u-boot.img programını isterler.

159

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*u-boot'un yüklenmesi*

net.net



- > tools/mkimage
Host tarafında çalışan, u-boot imajı üretme programıdır.
Genelde /usr/local/bin gibi bir yere kopyalanır.
Çok basit bir işleve sahiptir.
Herhangi bir dosyanın tepesine 64 baytlık bir alan ekler.
Dosyanın içeriğine hiç karışmaz.
Dosya ASCII dahi olabilir.

```
mkimage [-x] -A arch -O os -T type -C comp -a addr -e ep -n name
-d data_file[:data_file...] image
  -A ==> set architecture to 'arch'
  -O ==> set operating system to 'os'
  -T ==> set image type to 'type'
  -C ==> set compression type 'comp'
  -a ==> set load address to 'addr' (hex)
  -e ==> set entry point to 'ep' (hex)
  -n ==> set image name to 'name'
  -d ==> use image data from 'datafile'
  -x ==> set XIP (execute in place)
```

160

Gömülü Linux eğitimleri, kuruluşu ve desteği

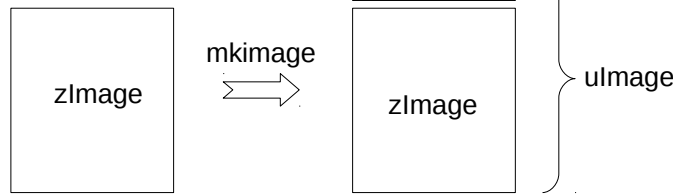
<http://ucanlinux.com>

u-boot'un yüklenmesi

net.net



```
$ mkimage -A arm
          -O linux
          -T kernel
          -C none
          -a 0x30008000
          -e 0x30008000
          -d arch/arm/boot/zImage
          /tftpboot/uImage
```



161

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*u-boot'un yüklenmesi*

net.net



u-boot/include/image.h

Image Header Magic Number	
Image Header CRC Checksum	
Image Creation Timestamp	
Image Data Size	
Data Load Address	-a
Entry Point Address	-e
Image Data CRC Checksum	
Operating System	-O
CPU architecture	-A
Image Type	-T
Compression Type	-C
Image Name	-n

zImage

162

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot'un yüklenmesi

net.net



- > Genelde u-boot seri kanal üzerinden RAM belleğe yüklenir. Sonra RAM bellekteki u-boot.bin programı NAND bölümüne yazılır.
- > Bu tekniğin tam uygulanması için mutlaka el kitaplarına bakılmalıdır.
- > ROM boot tarafından açılış yapılır.
 - > loady
 - Ctrl+S, ymodem, u-boot.bin seç
 - > write to nand
 - Reboot

Sistem u-boot'dan açılır.
- > Buradaki şlemler mealen verilmiştir, tamamen borda bağımlılık vardır.

163

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*u-boot'un yüklenmesi*

net.net



- > Bir diğer genel uygulamada ise MLO ve u-boot.bin programları MMC'nin 1. bölümüne kopyalanır.
- > 1. bölüme genelde vfat dosya sistemi kurulur.
- > Sector size, number of heads, partion type, boot flag, number of cylinders, starting sector number gibi pek çok değişkene bağımlılık olabilir.
- > Açılışta MMC'den açılması için bazı dip switch'lerin ayarlanıyor olması gerekebilir.
- > Sonuç olarak ROM boot loader'dan açılış yapıp u-boot'un yüklenmesi tamamı ile borda bağlı olup, el kitaplarından bakılmalıdır.

164

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

u-boot'un yüklenmesi

net.net



- > Aşağıda, mini2440 makinesine u-boot'un ilk yüklenmesi için örnek verilmiştir.
- > Bu örnek başka hiç bir makinede geçerli olmayabilir.
- > Bu tür ilk yüklemeler tamamen borda özgüdür.
- > u-boot.bin programı seri kanal yardımı ile NAND'a yazılmaktadır.
- > u-boot.bin programının elde edilmiş olduğunu kabul ediyoruz.

165

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*u-boot'un yüklenmesi*

net.net



```
$ minicom -s -w -c on
115200, 8n1, noflow tanımını yap.
```

S2 anahtarını NOR konumuna getir.

```
$ dd if=u-boot.bin of=u-boot-nand2k.bin bs=2k conv=sync
```

```
Supervivi> load flash 0 241664 u
```

```
$ smdk-usbd -f u-boot-nand2k.bin
```

- > Burada fiziksel belleğin tam başına, 0x30000000 adresine u-boot-nand2k.bin yüklenir.
- > 0, nand flash başlangıç adresidir, 241,664 sayısı ise u-boot-nand2k.bin dosyasının boyudur. u değeri ise seri protokolün adıdır.
- > Cihaz kapatılır, S2 anahtarı NAND konumuna getirilir, cihaz açılır ve u-boot promptu gelir.

166

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot'un Derlenmesi

net.net



- › Her bord üreticisi u-boot üzerinde borda uygun güncellemeleri yapar.
- › Her bord için bir config dosyası bulunur. Bu dosyanın ismi el kitaplarından veya borda ait örnek betiklerden veya boards.cfg dosyasından elde edilmelidir.
- › u-boot, kernel ve buildroot için önerilen toolchain kullanılmalıdır.
- › Bazı bordlar için u-boot, kernel ve br için farklı toolchain'ler önerilebilir, bu öneriye sadık kalınmalıdır.
- › Özellikle toolchain için önerilen sürüm numaralarına dikkat edilmelidir. Aynı derleyicinin farklı sürümleri dahi sorun yaratmaktadır.
- › Sonraki örnek min2440 için verilmiştir ama tamamen geneldir, her borda uygulanabilir. Config isimi ve git sunucusunun farklı olacağı aşikardır.

167

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot'un Derlenmesi

net.net



```

$ export PROJECT_HOME=/gomsis/mini2440

$ export UBOOT_SRC=$PROJECT_HOME/src/u-boot
$ export UBOOT_OUT=$PROJECT_HOME/out/u-boot

$ export CROSS="arm-none-linux-gnueabi-"

$ cd $PROJECT_HOME/src
$ git clone git://repo.or.cz/u-boot-openmoko/mini2440.git u-boot

$ make -C $UBOOT_SRC O=$UBOOT_OUT mini2440_config

$ make -C $UBOOT_SRC O=$UBOOT_OUT CROSS_COMPILE=$CROSS -j2

$ cp $UBOOT_OUT/tools/mkimage /usr/local/bin
$ cp $UBOOT_OUT/u-boot.bin /tftpboot

```

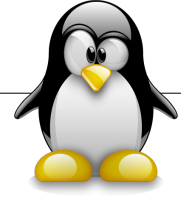
168

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Çekirdeğin Derlenmesi

net.net



```
$ make ARCH=arm mini2440_defconfig
$ make ARCH=arm menuconfig
$ make ARCH=arm CROSS_COMPILE=$CROSS -j2
$ mkimage -A arm -O linux -T kernel -C none
  -a 0x30008000 -e 0x30008000
  -d arch/arm/boot/zImage /tftpboot/uImage

veya

$ make ARCH=arm CROSS_COMPILE=$CROSS -j2 uImage

$ make ARCH=arm CROSS_COMPILE=$CROSS modules

# -C ve O= eklenmelidir.
```

169

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kök Dosya Sisteminin Kurulması

net.net



- > Çekirdeğin bağladığı ilk dosya sistemine kök dosya sistemi (root file system, RootFS) denir.
 - > RootFS pek çok ortamda oturabilir. RootFS'in farklı ortamlarda oturması kuruluş tekniğini değiştirmez.
 - > Ufak projeler için RootFS, busybox ile kurulabilir.
 - > Daha büyük projeler için br kullanılabilir.
 - > Hem busybox hem de br sisteminde prensip çok basittir.
- Örnek bir iskelet RootFS sistemi, hali hazırda mevcuttur. Sonra derlenen programlar bu iskelet RootFS içine dağıtılır. Bu RootFS içinde hem açılış betikleri hem de gerekli kütüphaneler bulunur. Üretilen RootFS sonra el yordamı ile güncellenir.
- > Şimdilik busybox destekli sistem kurulacak, ilerleyen bölümlerde br sistemine geçilecektir.

170

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kök Dosya Sisteminin Kurulması

net.net



- > busybox sistemi tar.gz gibi bir bütün olarak indirilmemelidir.
- > Aşağıdaki gibi git sisteminin kullanılması tavsiye edilir.


```
$ git clone git://busybox.net/busybox.git
$ git branch -a
$ git checkout remotes/origin/1_NN_stable
$ git branch
```
- > İlerleyen zamanlarda sadece \$ git pull yazarak son güncellemeler elde edilir.
- > O= derleme tekniği ile mevcut kaynak kodu sabit kalır. Böylece farklı derleyiciler veya farklı busybox sürümleri tek bir kaynaktan, birbirlerine karşımadan derlenebilirler.

171

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kök Dosya Sisteminin Kurulması

net.net



- ```
$ make -C $BUSYBOX_SRC O=$BUSYBOX_OUT clean
$ make -C $BUSYBOX_SRC O=$BUSYBOX_OUT menuconfig
$ make -C $BUSYBOX_SRC O=$BUSYBOX_OUT -j2
$ make -C $BUSYBOX_SRC O=$BUSYBOX_OUT install
```
- > Sembolik link kullanılması tavsiye edilir.
  - > Tool chain menuconfig sırasında tanımlanabilir. Çevre değişkeni olarak da verilebilir.
  - > \_install dizini incelenmelidir ki bütün busybox mantığı burada yatmaktadır.
  - > /bin, /sbin ve /usr dizinleri otomatik olarak kurulur. Ya diğer dizinler ve açılış betikleri?

172

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



## Kök Dosya Sisteminin Kurulması

net.net



```
$ cd /tftpboot/RootFS
$ rm -fr *
$ cp -a /gomsis/mini2440/RootFS.skel/* .
$ cp -a $BUSYBOX_OUT/_install/* .
$ chown -R root:root *
$ chmod +s bin/busybox
$ rm linuxrc
```

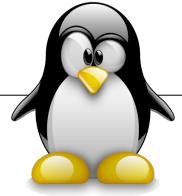
- RootFS.skel/ altında örnek bir kök dosya sistemi vardır. İçinde /bin, /sbin ve /usr yoktur.
- Eksik olan /bin, /sbin ve /usr dosyaları busybox'tan, diğer bütün dosyalar RootFS.skel'den /tftpboot/RootFS altına kopyalanır.
- Kök dosya sistemi /tftpboot/RootFS altında, ağ üzerinden kullanıma hazırdır.

173

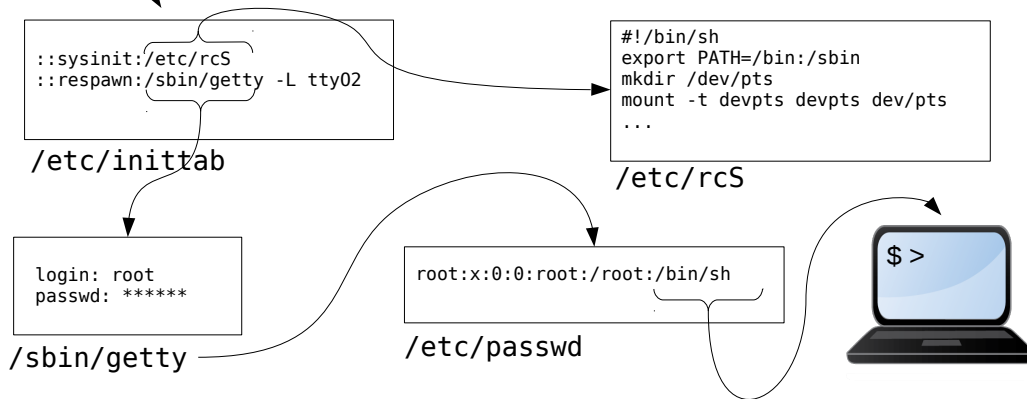
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## Açılış Sırası



Second Stage Boot Loader tarafından çekirdek belleğe yüklenir. Çekirdek kök dosya sistemini mount eder ve /sbin/init programını başlatır.



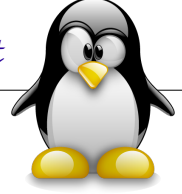
174

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## Açılış Betiği

net.net



- > Açılış betiği RootFS/etc/rcS içindedir.
- > Çok basit ama işleyen bir rcS dosyası aşağıda verilmiştir.
- > Çekirdek açıldıktan sonra NFS üzerinden RootFS'i bağlar.
- > Hemen /sbin/init programını işletir.
- > /etc/inittab içinde bulunan sysinit satırı sayesinde rcS işletilir.
- > rcS bütün sistemi ayağa kaldırır.
- > Çok basit gibi gözükmesine rağmen mantık silsilesi en karmaşık sistemlerde bile neredeyse aynı kalmaktadır.

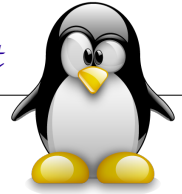
175

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## Açılış Betiği

net.net



```

/etc/rcS dosyası:

#!/bin/sh +x
#
export PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin

mount -t sysfs sysfs /sys
mount -t proc proc /proc

mkdir /dev/pts
mount -t devpts devpts /dev/pts

mkdir /dev/shm
mount -t tmpfs tmpfs /dev/shm

hostname UcanLinux

syslogd
klogd

ifconfig lo 127.0.0.1 up
route add -net 127.0.0.0 netmask 255.0.0.0 gw 127.0.0.1 lo

telnetd
df

```

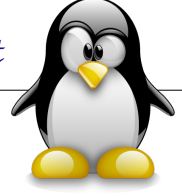
176

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## Açılış Betiği

net.net



/etc/inittab dosyası:

```
::sysinit:/etc/rcS
::respawn:/sbin/getty -L ttySAC0 115200 vt100
::shutdown:/bin/sync
::shutdown:/bin/umount -a -r
```

177

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## Açılış Betiği

net.net



- Bu betikte /dev sistemi çekirdek tarafından otomatik olarak bağlanmaktadır.
- Bunun için gerekli destek aşağıdaki gibi verilmelidir.

```
Device Drivers --->
 Generic Driver Options --->
 [*] Create a kernel maintained /dev tmpfs
 [*] Automount devtmpfs at /dev
```

- /dev dosya sistemi başka biçimlerde de kurulabilir.

```
/etc/rcS'de "mdev -s" ile,
cp -a /tftpboot/dev/* /dev ile,
dev.tar.gz'yi açarak,
mknod ile anında kurarak,
...
```

178

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Test**net.net*

- > BORD u-boot ile açılır ve aşağıdaki gibi tftp+NFS üzerinden test edilebilir.
- > Önce çekirdek yükle...
 

```

> ping 10.0.0.4 # server bağlı mı?
> bdfinfo # Adres başlangıcının yarısını al.
> tftp 32000000 uImage # Yükle.
> iminfo # Doğru gelmiş mi?
> bootm # Başla ve panikle!

```
- > Kök dosya sistemi NFS üzerinden bağlanacaktır. Host tarafında tftp ve NFS sunucularının çalıştığı ve düzgün ayarlandığı kabul edilmektedir.

179

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Test**net.net*

- > RootFS bağla ve login'e gel.
 

```

u-boot> setenv bootargs console=ttySAC0,115200
ip=10.0.0.111:10.0.0.4:10.0.0.4:255.255.255.0:test1:eth0:off
root=/dev/nfs
rw
nfsroot=10.0.0.4:/tftpboot/RootFS

u-boot> tftp 32000000 uImage

u-boot> bootm

```
- > Autoboot için, yukarıda el yordamı ile yazılan satırlar aralarına ';' veya && konularak bootcmd değişkenine atanır.
- > u-boot, bootcmd içindeki bütün komutları gözü kapalı işletir. Böylece otomatik açılış gerçekleşir.
- > bootargs doğrudan çekirdeğe geçer.

180

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*tftp**net.net*

> host: /etc/xinetd.d/tftp

```
service tftp
{
 protocol = udp
 port = 69
 socket_type = dgram
 wait = yes
 user = nobody
 server = /usr/sbin/in.tftpd
 server_args = /tftpboot
 disable = no
}
```

181

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*NFS**net.net*

> host: /etc/exports

```
/tftpboot 10.0.0.111(rw,insecure,no_subtree_check,async,no_root_squash)
```

```
$ exportfs -avr
```

182

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*eth0**net.net*

```
/etc/network/interfaces:
```

```
iface eth0 inet static
 address 10.0.0.4
 netmask 255.255.255.0
```

```
$ ifconfig eth0
```

```
eth0 Link encap:Ethernet HWaddr 00:90:f5:dc:51:d2
 inet addr:10.0.0.4 Bcast:10.0.0.255 Mask:255.255.255.0
 UP BROADCAST MULTICAST MTU:1500 Metric:1
 RX packets:0 errors:0 dropped:0 overruns:0 frame:0
 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

183

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Login Ekranı**net.net*

```
TFTP from server 10.0.0.4; our IP address is 10.0.0.111
Filename 'uImage'.
Load address: 0x32000000
```

```
Bytes transferred = 2138588 (20a1dc hex)
Booting kernel from Legacy Image at 32000000 ...
 Image Name:
 Created: 2014-04-12 5:41:12 UTC
 Image Type: ARM Linux Kernel Image (uncompressed)
 Data Size: 2138524 Bytes = 2 MB
 Load Address: 30008000
 Entry Point: 30008000
 Verifying Checksum ... OK
 Loading Kernel Image ... OK
```

OK

184

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Login Ekranı**net.net*

```
Starting kernel ...

Uncompressing Linux.....
..... done, booting the kernel.
Linux version 2.6.32-rc8-12.nisan.2104-glebb861 (nazim@nkoc)
 (gcc version 4.2.3 (Sourcery G++ Lite 2008q1-126))
 #3 Sat Apr 12 08:41:08 EEST 2014
CPU: ARM920T [41129200] revision 0 (ARMv4T), cr=c0007177
...
IP-Config: Complete:
 device=eth0, addr=10.0.0.111, mask=255.255.255.0, gw=10.0.0.4,
 host=test1, domain=, nis-domain=(none),
 bootserver=10.0.0.4, rootserver=10.0.0.4, rootpath=

VFS: Mounted root (nfs filesystem) on device 0:13.
devtmpfs: mounted
Freeing init memory: 140K
```

185

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Login Ekranı**net.net*

```
UcanLinux
```

Welcome to UcanLinux  
<http://ucanlinux.com>

buildroot login: root

| Filesystem | Size  | Used  | Available | Use% | Mounted on |
|------------|-------|-------|-----------|------|------------|
| /dev/root  | 62.9G | 10.1G | 49.5G     | 17%  | /          |
| devtmpfs   | 29.5M | 0     | 29.5M     | 0%   | /dev       |
| tmpfs      | 29.6M | 0     | 29.6M     | 0%   | /dev/shm   |
| tmpfs      | 29.6M | 60.0K | 29.5M     | 0%   | /tmp       |

```
eth0 Link encap:Ethernet HWaddr 08:08:11:18:12:27
 inet addr:10.0.0.111 Bcast:10.0.0.255 Mask:255.255.255.0
```

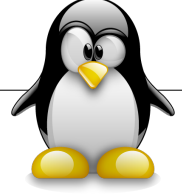
```
Linux buildroot 2.6.32-rc8-12.nisan.2104-glebb861 #3 Sat Apr 12 08:41:08 EEST
2014 armv4tl GNU/Linux
```

root@buildroot:~ #

186

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*nand.nand*

## Müşteri Sistemi

Çekirdek: NAND üzerinde.  
RootFS : NAND üzerinde

u-boot ve env, genelde NAND üzerindedir.

187

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*ubi/ubifs**nand.nand*

- u-boot, çekirdek ve kök dosya sistemi NAND üzerindedir.
- Eğer bord'un NAND desteği varsa, müşteriye verilecek cihaz için en ideal durumdur.
- ubi/ubifs sistemi kapanmaya karşı çok dayanıklıdır.
- Bu dayanıklılık dosya sistemi bazındadır. Diğer bir deyişle bir sonraki mount garantilenmiştir. Fakat dosya güncelleme konusunda bir garanti sunmaz.
- net.net sisteminde hazırlanan u-boot, kernel ve RootFS kullanılacaktır.
- Burada yapılacak esas iş, NAND üzerinde RootFS'in oturacağı ubifs sistemi kurmaktır.

188

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



*ubi/ubifs**nand.nand*

- Piyasada satılan NAND destekli bütün bordlar genelde bu şekilde satılırlar. Yani u-boot, kernel ve RootFS NAND üzerindedir.
- Daha önce elde edilmiş, u-boot, kernel ve RootFS burada kullanılacaktır.
- Kuruluş tekniği çok basittir.

Öncelikle bord, net.net ile açılır.  
 u-boot ve env NAND'a raw yazılır.  
 uImage NAND'a raw yazılır.  
 RootFS için ubifs kurulur ve mount edilir.  
 Mevcut RootFS dizini kopyalanır.  
 u-boot değişkenleri atanır.

189

Gömülü Linux eğitimleri, kuruluşu ve desteği

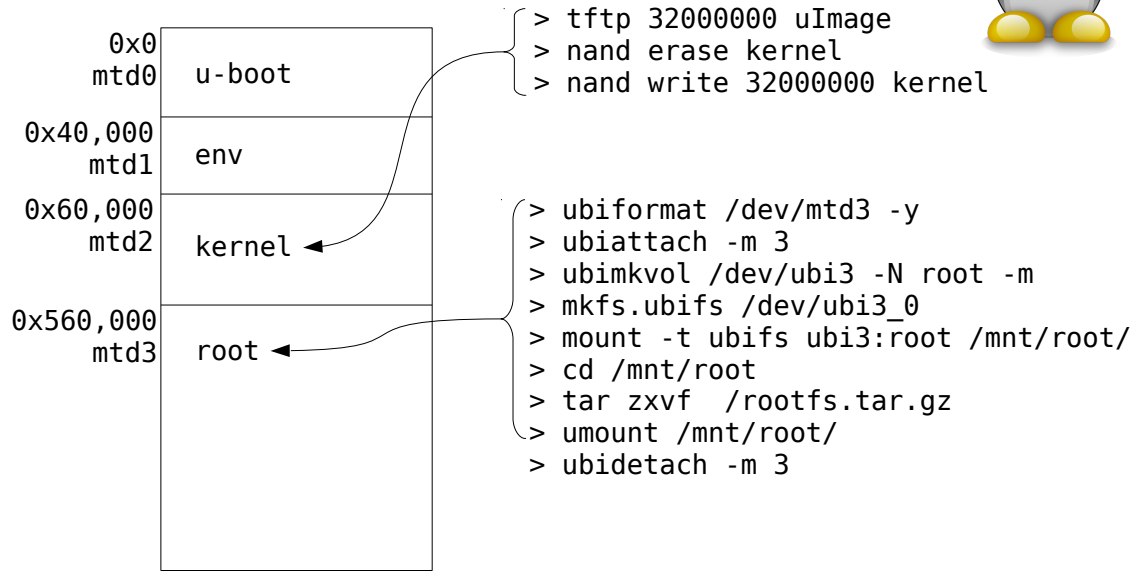
<http://ucanlinux.com>*Bölümlendirme**nand.nand*

- Sonraki sayfada verildiği gibi bir bölümlendirmenin mevcut olduğunu kabul edelim.
- Bu bölümlendirme mtdparts değişkeni ile tanımlanır. mtdparts bilgisi, bootargs yardımı ile çekirdeğe aktarılır. Böylece hem u-boot hem de çekirdek aynı bölümlendirme bilgisine sahip olacaktır.
- Genelde ilk 4 bölüm hep aynıdır, u-boot, env, kernel ve rootfs bölümlerini barındırır.
- İlk 3 bölüm her zaman raw olarak kullanılır. Yani bölüm içinde dosya sistemi bulunmaz.

190

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Bölümlendirme**nand.nand*

```
mtdparts=nand0:256k@0(u-boot),128k(env),5m(kernel),-(root)
```

191

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Çekirdeğin Kazınması**nand.nand*

```
MINI2440 # mtdparts
```

```
device nand0 <mini2440-nand>, # parts = 4
#: name size offset
mask_flags
0: u-boot 0x00040000 0x00000000 0
1: env 0x00020000 0x00040000 0
2: kernel 0x00500000 0x00060000 0
3: root 0x3faa0000 0x00560000 0
```

```
MINI2440 # tftp 32000000 uImage
Bytes transferred = 2138532 (20a1a4 hex)
```

```
MINI2440 # nand erase kernel
MINI2440 # nand write 32000000 kernel
MINI2440 # reset
```

192

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## Çekirdeğin Kazınması

nand.nand



NAND ile çalışırken bazı temel kurallara mutlaka uymak gerekir.

- Yazma işlemi 1'leri 0 yapabilir, tersini yapamaz. Bundan dolayı yazma işlemi yapmadan evvel bütün NAND bölgesini 1'ler ile doldurmak gerekir. nand erase işlemi bütün NAND bitlerini 1 yapar. Daha sonra nand write işlemi, 1'leri 0 yapar.
- Bütün adresler mutlaka PEB'in tam katları olmalıdır. PEB, genelde 128KiB değerindedir.
- Bütün dosya boyları mutlaka sektör boyunun katları olmalıdır. Sektör boyları veya minimum I/O size değeri genelde 2048'dir.

193

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

## RootFS'in Kuruluşu

nand.nand



- RootFS birçok teknik ile kurulabilir. ubinize, updatevol, ubiformat gibi.
- Burada verilen yöntem başlangıç veya ilk kuruluş için idealdir.
- /tftpboot/RootFS dizininin rootfs.tar.gz şeklinde paketlenmiş ve /tftpboot/RootFS/ altına kopyalandığını kabul edelim.

194

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*RootFS'in Kuruluşu**nand.nand*

```
ubiformat /dev/mtd3 -y
ubiattach -m 3
ubimkvol /dev/ubi3 -N root -m
mkfs.ubifs /dev/ubi3_0
mount -t ubifs /dev/ubi3_0 /mnt/root/ # veya
mount -t ubifs ubi3:root /mnt/root/

df /mnt/root
Filesystem Size Used Available Use% Mounted on
ubi3:root 931.3M 16.0K 931.3M 0% /mnt/root

cd /mnt/root

tar zxvf /rootfs.tar.gz

umount /mnt/root/
UBIFS: un-mount UBI device 3, volume 0

root@buildroot:~ # ubidetach -m 3
UBI: mtd3 is detached from ubi3
```

195

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Test**nand.nand*

```
> setenv bootargs console=ttySAC0,115200
ubi.mtd=3
rootfstype=ubifs
root=ubi0:root

> nboot kernel

> bootm
```

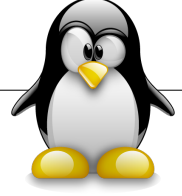
Autoboot için:

```
> setenv bootcmd 'setenv bootargs console=ttySAC0,115200 ubi.mtd=3
rootfstype=ubifs root=ubi0:root ; nboot kernel ; bootm'
> saveenv
> reset
```

196

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*mmc.mmc*

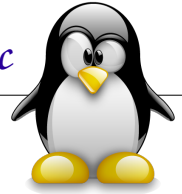
Test/İlk Açılış Sistemi

Çekirdek: MMC üzerinde.  
RootFS : MMC üzerinde

u-boot ve env, genelde MMC, 1. bölüm üzerindedir.

197

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Bölümlendirme**mmc.mmc*

- U-Boot, env, çekirdek ve RootFS MMC üzerindedir.
- Müşteriye verilecek sistemlerde bu tür bir düzenleme tavsiye edilmez.  
MMC kırılmalıdır, sürekli yazmada ömürleri kısadır, ubifs gibi desteği yoktur.
- MMC daha çok test sistemlerinde, ilk açılışlar için veya log tutmak için kullanılabilir.
- MMC açılışları için ek başına u-boot yeterli değildir.  
Bundan dolayı MLO gibi, u-boot sisteminin daha basit bir hali ile açılış yapılır.  
MLO, daha sonra u-boot'u yükler.
- Genelde MMC'nin 1. bölümüne vfat kurulur ve önce MLO kopyalanır.

198

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Bölümlendirme**mmc.mmc*

- Kök dosya sisteminin R0 bağlanması tavsiye edilir. Bu durumda ext2 dosya sistemi kullanılabilir.
- Kök dosya sistemi RW bağlanacaksa ext4 gibi, kapanmaya dayanıklı, log tabanlı bir dosya sistemi kullanılmalıdır.
- MMC üzerinde ASLA swap kullanılmamalıdır.
- MMC kartı host'a takılarak kopyalama işlemleri yapılabilir.
- Ya da bord net.net üzerinden açılır ve kopyalamalar NFS üzerinden yapılabilir.
- uEnv.txt dosyası içinde uenvcmd değişkeni varsa, run ile işletilir. Bu özelliğe, MMC Autoboot olarak bakılabilir.

199

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Bölümlendirme**mmc.mmc*

| mmcblk0    |                                             |
|------------|---------------------------------------------|
| p1<br>vfat | MLO<br>u-boot.bin<br>uImage<br>uEnv.txt     |
| p2<br>ext4 | /bin<br>/etc<br>/lib<br>/tmp<br>/var<br>... |

```
$ fdisk /dev/mmcblk0
$ mkfs.vfat /dev/mmcblk0p1
$ mkfs.ext4 /dev/mmcblk0p2
```

200

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Test**mmc.mmc*

```
> setenv bootargs console=ttySAC0,115200
rootfstype=ext4
root=/dev/mmcblk0p2
rw
rootdelay=2

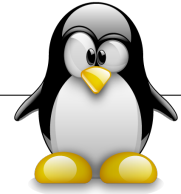
> mmcinit

> fatload mmc 0 32000000 uimage

> bootm
```

201

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*net.ramfs*

## Ufak RootFS Durumu

Çekirdek: Ağ üzerinde.  
RootFS : Çekirdeğe gömülü.

u-boot ve env, NAND veya MMC üzerinde olabilir.

202

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Çekirdek**net.ramfs*

- Bu yöntemde sadece çekirdek mevcuttur. Kök dosya sistemi çekirdek içine gömülmüştür.
- Test için uygun değildir, çünkü her değişiklikten sonra çekirdeğin derlenmesi gerekir.
- Çekirdek derlemesi sırasında kök dosya sistemi çekirdek koduna eklenir.
- Eğer RootFS'in boyu ufaksa, müşteriye verilecek ideal bir sistemdir.
- net.ramfs yönteminde çekirdek ağ üzerinden yüklenecektir. Çekirdek yüklendiği ana RootFS de yüklenmiş demektir. Açılış inanılmaz hızlıdır.
- Ayrıca emergency sistemler için de bu teknik kullanılır. Önce ramfs açılır, işi bittikten sonra esas sistem açılabilir.

203

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Çekirdek**net.ramfs*

General setup ---&gt;

- ```
[*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
(/tftpboot/RootFS) Initramfs source file(s)

(1000) User ID to map to 0 (user root)
(1000) Group ID to map to 0 (group root)

[*] Support initial ramdisks compressed using gzip
```

204

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*RootFS**net.ramfs*

- Daha önce yapıldığı gibi, busybox derlenir. Elde edilen /bin, /sbin, /usr dizinleri iskelet RootFS ile birleştirilir ve /tftpboot altına RootFS ismi ile kopyalanır.
- Diğer örnek RootFS'den farklı olarak bütün dosya ve dizinlerin sahibi ve grubu, root:root değil 1000:1000 numaralı kullanıcıdır.
- Yapılan her değişiklikte çekirdek yeniden derlenmelidir.
- RootFS için otomatik olarak cpio sistemi kurulur, sıkıştırılır ve çekirdeğin altına eklenir.
- Çekirdek boyu büyüdüğünden, NAND veya oturacağı yer için yeterli yer olmalıdır.
- Dosya sistemi büyükse, örneğin /lib'ler ayrı bir NAND bölümünden mount edilebilir.

205

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*RootFS**net.ramfs*

- Bu yöntemde devtmpfs otomatik olarak bağlanmaz. Bundan dolayı mdev gibi bir yöntemle /dev kurulmalıdır. mdev'den önce /proc ve /sys mevcut ve bağlı olmalıdır.
- Çekirdek açılır açılmaz, cpio arşivini tmpfs destekli bir kök dosya sistemine açar ve bu dosya sistemini /dev/root'a bağlar.
- Bağlama işinden hemen sonra /init programını başlatır.
- /init programı çalışabilir her program veya betik olabilir.
- Genelde init->/sbin/init yapılır.
- Test amacı ile init->/bin/sh yapılabilir. Ama bu durumda line discipline mevcut değildir. Terminal değil, konsol gibi çalışılır.

206

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Açılış Betiği**net.ramfs*

```
#!/bin/sh +x
#
export PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin

mount -t proc proc /proc
mount -t sysfs sysfs /sys
mount -t devpts devpts /dev/pts
mount -t tmpfs tmpfs /dev/shm

echo /sbin/mdev >/proc/sys/kernel/hotplug
mdev -s

hostname UcanLinux

syslogd
klogd

ifconfig lo 127.0.0.1 up
route add -net 127.0.0.0 netmask 255.0.0.0 gw 127.0.0.1 lo

ifconfig eth0 10.0.0.111 netmask 255.255.255.0 up

telnetd
df
```

207

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Test**net.ramfs*

```
> setenv ipaddr 10.0.0.111
> setenv serverip 10.0.0.4
> setenv netmask 255.255.255.0 } = > set autoload no
                                     > set autostart no
                                     > dhcp veya
                                     > bootp

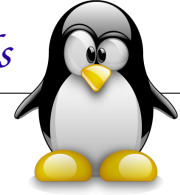
> tftp 32000000 uImage } = > set autoload yes

> bootm } = > set autostart yes
```

208

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*bootp/dhcp**net.ramfs*

```
$ dhcpd eth0 -f -d
```

```
Internet Systems Consortium DHCP Server 4.2.4
Copyright 2004-2012 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Wrote 0 deleted host decls to leases file.
Wrote 0 new dynamic host decls to leases file.
Wrote 0 leases to leases file.
Listening on LPF/eth0/00:90:f5:dc:51:d2/10.0.0.0/24
Sending on   LPF/eth0/00:90:f5:dc:51:d2/10.0.0.0/24
Sending on   Socket/fallback/fallback-net

DHCPDISCOVER from 08:08:11:18:12:27 via eth0
DHCPOFFER on 10.0.0.113 to 08:08:11:18:12:27 via eth0
DHCPREQUEST for 10.0.0.113 (10.0.0.4) from 08:08:11:18:12:27 via eth0
DHCPACK on 10.0.0.113 to 08:08:11:18:12:27 via eth0
```

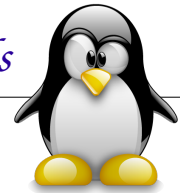
> Ya da aşağıdaki gibi başlat ve /var/log/syslog'a bak.

```
$ service isc-dhcp-server start
```

> Ayrılan IP'ler için bak: /var/lib/dhcp/dhcpd.lease

209

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*bootp/dhcp**net.ramfs*

```
# /etc/dhcp/dhcpd.conf:
```

```
ddns-update-style none;
allow bootp;
allow booting;
```

```
subnet 192.168.1.0 netmask 255.255.255.0{
}
```

```
subnet 10.0.0.0 netmask 255.255.255.0{
}
```

```
host test1{
  hardware ethernet 08:08:11:18:12:27;
  fixed-address 10.0.0.114;
  next-server 10.0.0.4;           # tftp server adresi.
  filename "ulmage";           # "autoload yes" ise yüklenir.
}
```

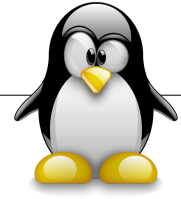
```
option host-name "test1";
option domain-name "ucanlinux.com";
option routers 10.0.0.4;
option root-path "/tftpboot";
}
```

210

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

nand.ramfs



Ufak RootFS Durumu

Çekirdek: NAND içinde.
RootFS : Çekirdeğe gömülü.

u-boot ve env, NAND veya MMC üzerinde olabilir.

211

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

mtddparts

nand.ramfs



- net.ramfs ile tamamen aynıdır.
Aynı yarar ve aynı sakıncalara sahiptir.
Farklı olarak çekirdek NAND üzerinden yüklenecektir.
- NAND bölümlendirmesi 2 türlü yapılabilir.
mtddparts parametresi çekirdeğe gönderilir, dinamik tanım.
Ya da kaynak kodu içinde doğrudan statik bir tanım yapılır.
- Statik tanım nihai sistemlerde, dinamik tanım test sistemlerinde yapılabilir.
- mini2440 için mtddparts değişkeni
arch/arm/mach-s3c2440/mach-mini2440.c içinde yapılabilir.
- C dosyası her bord için farklıdır.

212

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*mtdparts**nand.ramfs*

```
static struct mtd_partition mini2440_default_nand_part[] = {
    [0] = {
        .name      = "u-boot",
        .size = SZ_256K, /* 256K */
        .offset    = 0,
    },
    [1] = {
        .name      = "env", /* 128K */
        .size = SZ_128K,
        .offset    = SZ_256K,
    },
    [2] = {
        .name      = "kernel0",
        .size = 0x1000000, /* 16M */
        .offset    = SZ_256K + SZ_128K,
    },
    [3] = {
        .name      = "kernel1",
        .size = 0x1000000, /* 16M */
        .offset    = SZ_256K + SZ_128K + 0x1000000,
    },
    [4] = {
        .name      = "root",
        .size = 0x4000000, /* 64M */
        .offset    = SZ_256K + SZ_128K + 0x1000000 + 0x1000000,
    },
};
```

213

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*mtdparts**nand.ramfs*

```
[5] = {
    .name      = "app",
    .size = 0x4000000, /* 64M */
    .offset    = SZ_256K + SZ_128K + 0x1000000 + 0x1000000 + 0x4000000,
},
[6] = {
    .name      = "config",
    .size = 0x1000000, /* 16M */
    .offset    = SZ_256K + SZ_128K + 0x1000000 + 0x1000000 + 0x4000000 + 0x4000000,
},
[7] = {
    .name      = "logs",
    .size = 0x8000000, /* 128M */
    .offset    = SZ_256K + SZ_128K + 0x1000000 + 0x1000000 + 0x4000000 + 0x4000000 +
                0x1000000,
},
[8] = {
    .name      = "backup", /* kalan */
    .size = MTDPART_SIZ_FULL,
    .offset    = SZ_256K + SZ_128K + 0x1000000 + 0x1000000 + 0x4000000 + 0x4000000 +
                0x1000000 + 0x8000000,
},
};
```

214

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*mtdparts**nand.ramfs*

- › Aynı bölümlendirmenin dinamik tanımını,

```
> setenv mtdparts mtdparts=mini2440-nand:
    256k@0(u-boot),
    128k(env),
    16m(kernel0),
    16m(kernel1),
    64M(root),
    64M(app),
    16M(config),
    128M(logs),
    -(backup)
```

215

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*NAND'a Yazmak**nand.ramfs*

- › net.ramfs için kurulan ve içinde RootFS bulunan uImage dosyası hiç değiştirilmeden kullanılacaktır.
- › NAND'a yazılan her dosya page size değerinin tam katı olmalıdır.

```
$ cd /tftpboot
$ dd if=uImage of=uImage2k bs=2k conv=sync
```

```
$ ls -l uImage*
-rwxrwxrwx 1 nazim nazim 3090904 Nis 28 08:17 uImage
-rw-rw-r-- 1 nazim nazim 3092480 Nis 28 08:34 uImage2k
```

- › NAND'a kuruluş için u-boot seviyesinde açılış yapılır.

216

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*mtddparts**nand.ramfs*

```
> nand info
Device 0: NAND 1GiB 3,3V 8-bit, page size 2048, sector size 128 KiB
```

```
> tftp 32000000 uImage2k
Bytes transferred = 3092480 (2f3000 hex)
```

```
MINI2440 # mtdparts
```

```
device nand0 <mini2440-nand>, # parts = 9
#: name                size                offset              mask_flags
0: u-boot              0x00040000         0x00000000         0
1: env                 0x00020000         0x00040000         0
2: kernel0            0x01000000         0x00060000         0
3: kernel1            0x01000000         0x01060000         0
4: root               0x04000000         0x02060000         0
5: app                 0x04000000         0x06060000         0
6: config             0x01000000         0x0a060000         0
7: logs               0x08000000         0x0b060000         0
8: backup             0x2cfa0000         0x13060000         0
```

```
active partition: nand0,0 - (u-boot) 0x00040000 @ 0x00000000
```

217

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Test**nand.ramfs*

```
> nand erase kernel1
> nand write 32000000 kernel1 2f3000

> setenv bootcmd_initramfs_nand 'setenv bootargs
                                console=ttySAC0,115200 ;
                                nand kernel1 ;
                                bootm'
```

```
> setenv bootcmd run bootcmd_initramfs_nand
```

```
> saveenv
> reset
```

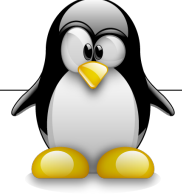
```
> Açıldıktan sonra incele:
```

```
$ cat /proc/mtd
$ cat /proc/partitions
$ cat /proc/cmdline
```

218

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

net.cpioNet

Ayrık Initramfs

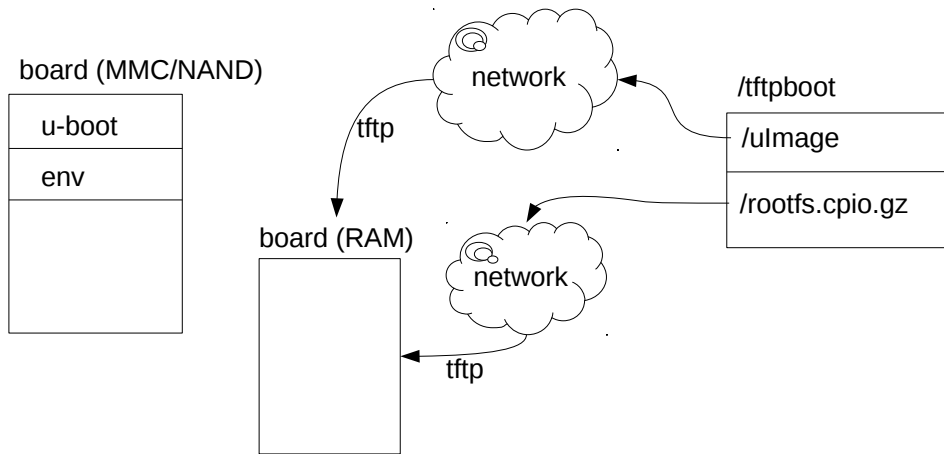
Çekirdek: Ağ üzerinde.

RootFS : Initramfs/Initrd Biçiminde ve ağ üzerinde.

u-boot ve env, NAND veya MMC üzerinde olabilir.

219

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*net.cpioNet*

220

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Giriş

net.cpioNet



- Kök dosya sistemi initramfs formatındadır fakat çekirdek içine gömülü değildir.
- RootFS, sıkıştırılmış cpio arşivi halinde, herhangi bir ortamda bulunabilir.
Bu örnek uygulamada arşiv dosyası ağ üzerindedir ve tftp ile yüklenecektir.
- net.ramfs örneğindeki RootFS sistemi, cpio arşivi haline getirilip sıkıştırılacak, sonra da u-boot imajı haline getirilecektir.
- RootFS çekirdeğe gömülü olmadığı için ufak olması gerekmez, ama yine de tmpfs kullandığı için, mevcut RAM'a göre çok büyük olmamalıdır.
- RootFS'deki programların GPL olması gerekmez.

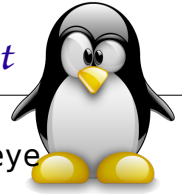
221

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Giriş

net.cpioNet



- RootFS güncellemesi çok kolaydır, çekirdek derlemeye gerek yoktur.
- Hem müşteriye verilecek son sistemde hem de test ortamında rahatlıkla kullanılabilir.

222

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Cpio Arşivi**net.cpioNet*

```

$ cd RootFS
$ sudo chown -R root:root *
$ find . | cpio -v -o -H newc | gzip > rootfs.cpio.gz
$ mkimage -A arm
          -T ramdisk
          -C none
          -n "cpio test imaji"
          -d rootfs.cpio.gz
            urootfs.cpio.gz

$ ls -l rootfs.cpio.gz urootfs.cpio.gz

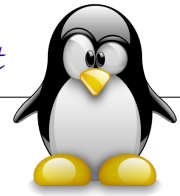
$ mkimage -l urootfs.cpio.gz
$ file urootfs.cpio.gz

$ cp urootfs.cpio.gz /tftpboot

```

223

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Test**net.cpioNet*

```

> tftp 32000000 uImage
> tftp 30800000 urootfs.cpio.gz
> iminfo 32000000
> iminfo 30800000
> bootm 32000000 30800000

> bootm'nin en genel yapısı:
  > bootm kernel initramfs/initrd fdt
    initramfs/initrd yoksa - yazılır.

```

224

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

nand.cpioNand

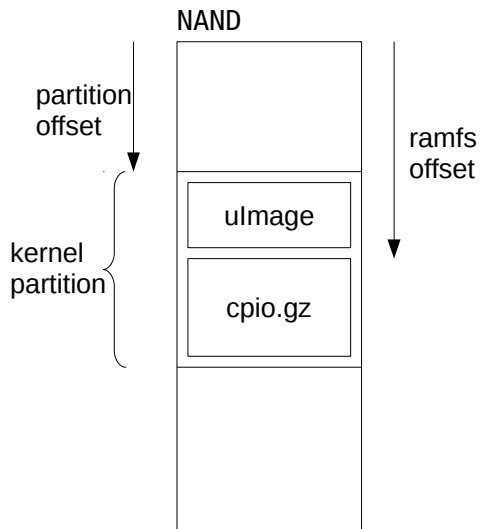
Ufak RootFS Durumu

Çekirdek: NAND içinde.
RootFS : NAND içinde.

u-boot ve env, NAND veya MMC üzerinde olabilir.

225

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*nand.cpioNand*

226

Gömülü Linux eğitimleri, kuruluşu ve desteği

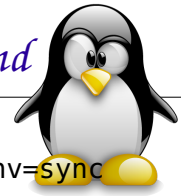
<http://ucanlinux.com>

*Giriş**nand.cpioNand*

- > Bu yöntemde hem çekirdek hem de RootFS sistemi NAND üzerindedir.
- > RootFS, cpio arşivi şeklindedir. Buradaki en büyük sorun arşivin NAND üzerinde nerede duracağıdır.
- > Arşiv için ayrı bir NAND bölümü yapılabilir ve nand write komutu ile arşiv bu bölüme yazılabilir. Fakat bu yöntemde ayrı bir NAND bölümü olmalıdır. Bu tekniğin uygulanması çok basittir.
- > Diğer yöntemde çekirdek ve arşiv aynı NAND bölümüne yazılır. Burada bir miktar adres hesabı yapmak gerekir.

227

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Kuruluş**nand.cpioNand*

```
$ dd if=/tftpboot/uImage of=/tftpboot/uImage2k bs=2048 conv=sync
$ IMAGE_SIZE=`stat -c "%s" /tftpboot/uImage2k`
$ RAMFS_SIZE=`stat -c "%s" /tftpboot/urootfs.cpio.gz`
$ RAMFS_OFF=$((17170432+$IMAGE_SIZE))
$ printf "kernel size: %d 0x%x\n" $IMAGE_SIZE $IMAGE_SIZE
$ printf "rootfs size: %d 0x%x\n" $RAMFS_SIZE $RAMFS_SIZE
$ printf "rootfs off : %d 0x%x\n" $RAMFS_OFF $RAMFS_OFF
$ cat /tftpboot/uImage2k /tftpboot/urootfs.cpio.gz >
  /tftpboot/nand_linux_initramfs
$ dd if=/tftpboot/nand_linux_initramfs
  of=/tftpboot/nand_linux_initramfs_2k bs=2048 conv=sync
$ FULL_SIZE=`stat -c "%s" /tftpboot/nand_linux_initramfs_2k`
```

228

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Kuruluş**nand.cpioNand*

```

echo Aşağıdaki gibi test et.
echo
printf "> tftp 32000000 nand_linux_initramfs_2k\n"
printf "> nand erase kernel1\n"
printf "> nand write 32000000 kernel1 0x%x\n" $FULL_SIZE
printf "> reset\n\n"
printf "> nand read 32000000 kernel1 0x%x\n" $IMAGE_SIZE
printf "> iminfo 32000000\n\n"
printf "> nand read 30800000 0x%x 0x%x\n" $RAMFS_OFF $RAMFS_SIZE
printf "> iminfo 30800000\n\n"
printf "> bootm 32000000 30800000\n\n"

```

229

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Test**nand.cpioNand*

```

# NAND'a kazımak.
> tftp 32000000 nand_linux_initramfs_2k

> nand erase kernel1
> nand write 32000000 kernel1 0x2f4000
> reset

# El yordamı ile açılış.

> nand read 32000000 kernel1 0x208800
> iminfo 32000000

> nand read 30800000 0x1268800 0xeb41f
> iminfo 30800000

> bootm 32000000 30800000

# Autoboot için yukarıdaki 5 komut bootcmd içine atılabilir.

```

230

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Giriş

mmc.cpioMMC



- > Bu yöntemde çekirdek ve cpio imajı MMC üzerindedir.
- > Bu tekniğin fazla bir özelliği yoktur, fakat güzel bir alıştırmadır.
- > Kuruluş son derece basittir.
MMC'nin 1. bölümüne vfat veya ext2 kurulur.
MLO, u-boot.bin, uImage ve cpio arşivi bu bölüme kopyalanır.

231

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kuruluş ve Test, ext2

mmc.cpioMMC

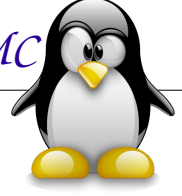


```
# Host tarafında:  
$ mkfs.ext2 /dev/mmcblk0p1  
$ cp MLO u-boot.bin uImage urootfs.cpio.gz /mnt/boot  
# MMC'yi cihaza tak.  
  
> mmcinit  
> ext2load mmc 0:1 32000000 uImage  
> ext2load mmc 0:1 30800000 urootfs.cpio.gz  
> bootm 32000000 30800000
```

232

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Kuruluş ve Test, vfat**mmc.cpioMMC*

```
# Host tarafında:

$ mkfs.vfat /dev/mmcblk0p1 -n BOOT

$ cp MLO u-boot.bin uImage urootfs.cpio.gz /mnt/boot

# MMC'yi cihaza tak.

> mmcinit
> fatls
> fatload mmc 0:1 32000000 uimage
> fatload mmc 0:1 30800000 urootfs.cpio.gz

> bootm 32000000 30800000
```

233

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot Teknikleri

234

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, autoscr



```
# test.scr
echo
echo bu bir deneme betidir.
echo MMC icinde ne var?
mmcinit
fatls mmc 0:1
echo
```

Host tarafında:

```
$ mkimage -T script -C none -n 'test betiği' -d test.scr /tftpboot/script.img
```

Bord tarafında:

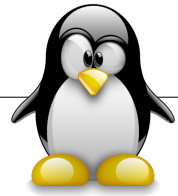
```
> tftp 32000000 script.img
> iminfo 32000000
> autoscr 32000000
```

235

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, base, memory display



Bütün adreslerin başına bu değer eklenir.
Varsayılan değer 0'dır.

```
> base 32000000
Base Address: 0x32000000
```

```
> tftp 0 script.img
> md.b 0 64
> md.w 0 64
> md.l 0 64
```

b: byte, 8 bit.
w: word, 16 bit
l: long word, 32 bit

236

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, bdfinfo



```
> bdfinfo
arch_number = 0x000007CF
env_t       = 0x00000000
boot_params = 0x30000100 <-- kernel parameters için.
DRAM_bank   = 0x00000000
-> start     = 0x30000000 <-- Çok önemli, +0x100 çekirdek
                        parametersi listesinin başlangıç
                        adresi.

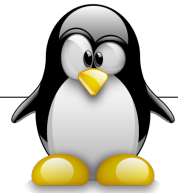
-> size      = 0x04000000 <-- 64M
ethaddr     = 08:08:11:18:12:27
ip_addr     = 10.0.0.111
baudrate    = 115200 bps
```

237

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*U-Boot, boot**



```
boot:
run bootcmd'nin kısa hali

bootd:
run bootcmd'nin başka bir kısa hali

bootm:

> bootm kernel
> bootm kernel initramfs
> bootm kernel initramfs fdt
> bootm kernel - fdt
```

238

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, bootp, dhcp



- > bootp # veya
- > dhcp

- > Her iki komut ile de ağ yapılandırması otomatik olarak yapılır.
- > Bu yapılandırma sonrasında, ağ ile ilgili env değişkenleri otomatik olarak atanır.
- > Eğer outoload değeri yes ise, tftp üzerinden boot dosyası yüklenir.
- > Eğer autostart yes ise yükleme sonrası otomatik olarak bootm işletilir.

239

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, compare, crc, coninfo



- > cmp a1 a2 0x32

- > a1 ve a2 adreslerini karşılaştırır.
İk farklı adres raporlanır.
cmp.w kabul edilir.

- > crc32 address count [addr]

- > CRC32 hesapla ve addr'ye sonucu yaz.

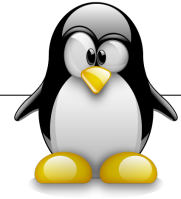
- > coninfo
serial 80000003 SIO stdin stdout stderr

- > S: system device, I: input, O:output

240

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, copy

```
> cp a b count

> bootp
Load address: 0x32000000

> iminfo 34000000
## Checking Image at 34000000 ...
Unknown image format!

> cp.b 32000000 34000000 2081a4
                        ^
                        çekirdeğin boyu

> iminfo 34000000
...
Verifying Checksum ... OK
```

241

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*U-Boot, dyn**

dynenv:
u-boot değişkenlerinin saklanacağı NAND cihazının başlangıç adresini tanımlar veya sorgular.

dynpart:
Her NAND bölümünün ofset değerini hesaplar.
U-Boot içinde gömülü olan bölümlendirme bilgisini kullanarak mtdparts değişkenini otomatik olarak atar.

242

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, go

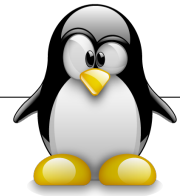
- > bare metal arm kodu işletmek içindir.
- > u-boot kaynak kodu içinde examples/ dizini altında hello_world.c mevcuttur, incelenebilir.
- > u-boot.bin programı da go ile başlatılabilir.

```
> tftp 32000000 hello.bin
> go 32000000
## Starting application at 0x32000000 ...
Hello World
Hit any key to exit ...

## Application terminated, rc = 0x0
```

243

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*U-Boot, itest*

```
> mw 200000 0

> while itest *200000 == 0
> do
>   tftp 200000 tqm5200/uImage
>   echo === done ====
> done

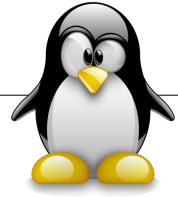
> check_ub_ver=if itest.s \${tmp} == \${ver};
   then
   echo equal;
   else
   echo diff ;
   fi

> run check_ub_ver
```

244

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, loop, memory display, memory test

```
loop:
> loop.l 32000000 0xF
```

16 adet long veriyi sonsuz kere okur.
Bir tür read testidir.
Ancak reset ile durur.

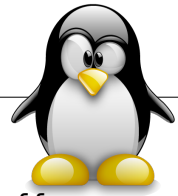
md: memory display

```
> tftp 32000000 script.img
> md.b 32000000 ff
> md.w 32000000 ff
> md.l 32000000 ff
```

mtest: simple memory test
NAND veya EPROM'a uygulanamaz.
RAM'ı değiştirerek test eder.
Çok uzun sürer.

245

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*U-Boot, memory write*

```
> md.b 32000000 10
32000000: da fd ff ff d9 fd ff ff d8 fd ff ff d7 fd ff ff
.....
```

```
> ? mw
mw [.b, .w, .l] address value [count]
- write memory
```

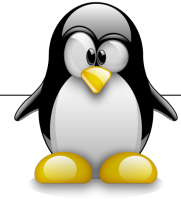
```
> mw.b 32000000 FF 10
```

```
> md.b 32000000 10
32000000: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
.....
```

246

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, memory modify

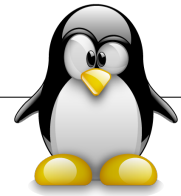
- > Adresin değerini gösterir.
Yeni veri bekler.
Boş enter yapılırsa veri değişmez.
Geçerli bir veri girilirse adrese atanır.
Geçersiz bir veri girilirse işlem biter.

```
> mm.b 32000000
32000000: 00 ? a
32000001: 01 ? b
32000002: 02 ? c
32000003: 03 ? .
> md.b 32000000 F
32000000: 0a 0b 0c 03 04 05 ff ff ff ff ff ff ff ff ....
>
```

- > nm'de adres artımı olmaz.
Daha çok device register testi içindir.

247

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*U-Boot, NFS*

```
> setenv bootargs console=ttySAC0,115200
> nfs 32000000 10.0.0.4:/tftpboot/uImage.initramfs veya
> nfs 32000000 /tftpboot/uImage.initramfs
> bootm
login gelir.
```

- > Eğer setenv autostart yes girilirse, bootm yapmaya gerek yoktur.

```
/etc/exports:
/tftpboot 10.0.0.114(rw,insecure,no_subtree_check,async,no_root_squash)
```

```
$ exportfs -avr
```

```
/etc/hosts:
10.0.0.114 mini2440
```

Bu tanım yoksa, "cannot umount" hatası veriyor!

248

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

U-Boot, rarpboot

> arp, MAC-IP eşlemesi yapar.

```
$ mkdir /var/lib/arpd
$ arpd eth0
$ arp -n -i eth0
```

> incomplete: ip var ama karşı gelen bir mac yok.
arp soru göndermiş ama kimse yanıt vermemiş.

> tcpdump ile inceleme:

```
$ tcpdump -i eth0
```

```
19:32:21.132675 ARP, Request who-has 10.0.0.111 tell nkoc.local, length 28
19:32:21.133068 ARP, Reply 10.0.0.111 is-at 08:08:11:18:12:27 (oui
Unknown), length 46
```

249

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*U-Boot, rarpboot*

```
19:32:26.129722 ARP, Request who-has nkoc.local tell 10.0.0.111, length 46
19:32:26.129778 ARP, Reply nkoc.local is-at 00:90:f5:dc:51:d2 (oui Unknown),
length 28
```

> ARP tablosuna 10.0.0.114'e karşı gelen IP girilmelidir.
Ki bord boot ederken kendi mac adresini yayınlayacak ve
buradan IP elde edecektir.

```
# arp -d 10.0.0.111 # Eskileri sil.
# arp -i eth0 -s 10.0.0.114 08:08:11:18:12:27
# arp -i eth0 -n
```

Test:

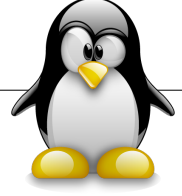
> rarpboot 32000000 /tftpboot/uImage.initramfs

> rarp ile adres alır, tftp ile yükleme yapar.

250

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



BuildRoot

251

Gömülü Linux eğitimleri, kuruluşu ve desteği

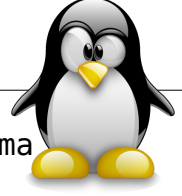
<http://ucanlinux.com>*BR*

- Tek başına busybox sistemi, gelişmiş bir gömülü sistem kurmak için yeterli olmayabilir.
- Bu durumda kurulacak her bir paket veya kütüphane, kaynak kodundan indirilmeli, ilgili mimari için yamaları geçilmeli, çapraz derlenmeli, RootFS içine install edilmeli, gerekli kütüphaneler install edilmeli, varsa açılış betikleri kurulmalıdır.
- Bir çok paket için bu işi yapabilmek son derece zahmetlidir.
- Bütün bu işleri otomatik olarak yapan programlar vardır. Şu anda bu programların içinde en yaygın olanı br'dir. br tamamen betiklerle yazılmıştır.

252

Gömülü Linux eğitimleri, kuruluşu ve desteği

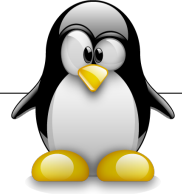
<http://ucanlinux.com>

BR

- BuildRoot veya kısaca br, Gömülü Linux Sistemi kurma aracıdır.
- Toolchain, RootFS, library, kernel ve bootloader imajı kurar.
- İstenilen paketler busybox veya kernel derlemesi gibi tek tek seçilebilir.
- Seçilen paketler, doğrudan kaynak kodundan otomatik olarak indirilir ve derlenir.
- Qt gibi paketlere sahip bir RootFS derlemek için peygamber sabrına gerek vardır.
- Her ne kadar kernel ve u-boot için de desteği olmasına rağmen bu desteklerin kullanılması tavsiye edilmez. Bordun el kitaplarından bakılarak u-boot ve linux için gerekli çalışma yapılmalıdır.

253

Gömülü Linux eğitimleri, kuruluşu ve desteği

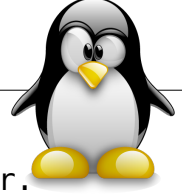
<http://ucanlinux.com>*BR*

- Pratikte sadece RootFS kurmak için kullanılır.
- Pek çok formatta RootFS kurulabilir. Örneğin doğrudan NAND'a yazılabilecek ubi imajı kurulabilir. Fakat bu tür kullanım çok katı olduğu ve güncellenmesi çok zor olduğu için tavsiye edilmez.
- RootFS sistemi rootfs.tar olarak kurulabilir. Bu durumda tar dosyası açılarak güncelleme yapılabilir.
- 700'den fazla pakete destek verir.
- Her 3 ayda bir kararlı sürüm yayınlanır.
- menuconfig ile paketler ve diğer özellikler seçilir. Nihayetide .config dosyası üretilir.

254

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

BR

- make ile derleme yapılır.
Gece yatmadan önce derleme verilmesi tavsiye edilir.
- Kaynak kodları dl/ dizini altına indirilir.
- Nihai imajlar output/images altında kurulur.
Şu anda sadece rootfs.tar ile ilgilenmekteyiz.
- rootfs.tar dosyası uygun bir dizine açılır ve gerekli güncellemeler yapılabilir.
- RootFS esas itibari ile mevcut bir iskelet RootFS üzerine kurulur.
İsterse kullanıcı kendi iskelet sistemini kurabilir.
- O= seçeneği kullanılmazsa, bütün çıkışlar kaynak kodu içinde output/ dizini altına yapılır.

255

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*BR*

- stages/ dizini çok önemlidir.
Host tarafında derleme yapılırken burada üretilen kütüphaneler kullanılacaktır.
- Mevcut bir toolchain kullanılabilir veya br'nin borda uygun bir toolchain üretmesi istenebilir.
- BR tarafından RootFS aşağıdaki gibi otomatik olarak üretilir.
output/target içine fs/skeleton içinde bulunan iskelet RootFS kopyalanır.
Gerekli kütüphaneler kopyalanır.
Paketler install edilir.
Varsa post-build betikleri işletilir.
ubifs, jffs2 gibi RootFS imajı üretilir.
- Çalışabilir bir RootFS elde edildikten sonra mutlaka .config dosyası ayrı bir yerde saklanmalıdır.

256

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

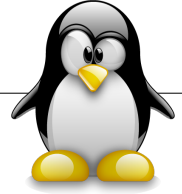
BR

› Aşağıdaki çalışma şekli tavsiye edilir.

- 1.busybox menuconfig ile önce busybox ayarlamaları yapılır.
- 2.menuconfig ile paketler seçilir.
- 3.Sabır çekilir ve make komutu verilir.
- 4.Elde edilen rootfs.tar komutu /tftpboot altına açılır.
- 5.RootFS/'i güncelleyen betikler yazılır.
- 6.Sonra amaca göre cpio arşivi, ubi imajı vs elde etmek için gerekli betikler yazılır.
- 7.Açılış testi başarılı olursa, kernel, busybox ve br için üretilmiş olan .config dosyaları ayrı bir yerde saklanmalıdır.

257

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*BBB*

Beagle
Bone
Black

Flatten Device Tree Örneği

258

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*U-Boot**BBB*

```

$ git clone git://git.denx.de/u-boot.git

# u-boot/boards.cfg satır1:
Active arm armv7 am33xx ti am335x am335x_boneblack
am335x_evm:SERIAL1,CONS_INDEX=1,EMMC_BOOT Tom Rini <trini@ti.com>

# u-boot/board/ti/am335x/README mutlaka okunmalıdır.
#
# configs:
# u-boot/include/configs/am335x_evm.h # Esas
# /ti_am335x_common.h
# /ti_armv7_common.h # Prompt burada.

$ make CROSS_COMPILE=${CROSS} -C $UBOOT_SRC O=$UBOOT_OUT
am335x_boneblack_config

$ make CROSS_COMPILE=${CROSS} -C $UBOOT_SRC O=$UBOOT_OUT -j2

$ ls -l $UBOOT_OUT/MLO
$ ls -l $UBOOT_OUT/u-boot.bin

```

259

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Kernel, Load**BBB*

- BBB çekirdeği FDT kullanır. FDT bilgisi, aynen initramfs gibi çekirdeğe de eklenebilir, ayrı da kullanılabilir. Kolaylık için her zaman çekirdeğin altına eklenmelidir.
- Çekirdek aşağıdaki gibi yüklenir.
- ARM makinelerinde genel 2.6.x veya 3.2.x çekirdekler kullanılır. 3.8 çekirdeği ARM işlemciler için çok yenidir.

```

$ git clone git://github.com/beagleboard/kernel.git
$ cd kernel
$ git checkout 3.8
$ ./patch.sh
$ cp configs/beaglebone kernel/arch/arm/configs/beaglebone_defconfig
$ wget http://arago-project.org/git/projects/?p=am33x-
cm3.git;a=blob_plain;f=bin/am335x-pm-firmware.bin;hb=HEAD
-O kernel/firmware/am335x-pm-firmware.bin

```

260

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kernel, menuconfig

BBB

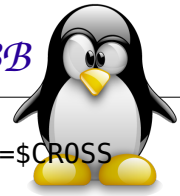
```
$ make -C $KERNEL_SRC O=$KERNEL_OUT ARCH=arm clean
$ make -C $KERNEL_SRC O=$KERNEL_OUT ARCH=arm beaglebone_defconfig
$ make -C $KERNEL_SRC O=$KERNEL_OUT ARCH=arm CROSS_COMPILE=$CROSS
  menuconfig
```

261

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Kernel, build

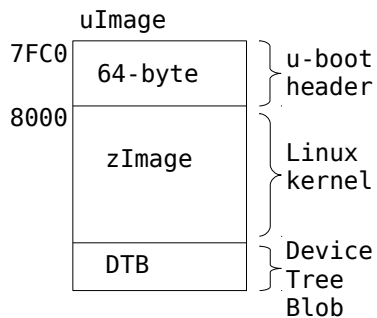
BBB

```
$ make -C $KERNEL_SRC O=$KERNEL_OUT ARCH=arm CROSS_COMPILE=$CROSS
  uImage dtbs -j2
$ make -C $KERNEL_SRC O=$KERNEL_OUT ARCH=arm CROSS_COMPILE=$CROSS
  uImage-dtb.am335x-boneblack
$ cp $KERNEL_OUT/arch/arm/boot/uImage-dtb.am335x-boneblack
  /tftpboot/uImage
```

262

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*Test, XIP**BBB*

```
> setenv bootargs console=tty00,115200n8
> setenv autoload no
> dhcp
> tftp 80007fc0 uImage
> bootm
```

veya

```
> setenv bootargs console=tty00,115200n8
> setenv autoload yes
> setenv autostart yes
> setenv loadaddr 80007fc0
> dhcp
```

263

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*RootFS, Busybox**BBB*

```
# clone

$ git clone git://busybox.net/busybox.git
$ git branch -a
$ git checkout remotes/origin/1_NN_stable
$ git branch

# menuconfig

$ make -C $BUSYBOX_SRC O=$BUSYBOX_OUT clean
$ make -C $BUSYBOX_SRC O=$BUSYBOX_OUT menuconfig

# build

$ make -C $BUSYBOX_SRC O=$BUSYBOX_OUT -j2
$ cd $BUSYBOX_OUT
$ ls -l busybox
$ file busybox
```

264

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

RootFS, Busybox

BBB



```
# install
$ make -C $BUSYBOX_SRC O=$BUSYBOX_OUT install
```

> install işlemi \$BUSYBOX_OUT/_install altına /bin, /sbin, /usr dizinlerini kurar. Ayrıca /linuxrc dosyası da vardır.

> /linuxrc dosyası, eski moda initrd sistemleri içindir. Artık initramfs kullanıldığı için /linuxrc dosyasına gerek yoktur. initramfs sistemi /init kullanılır. Bundan dolayı /init dosyası çalışabilir bir dosyayı göstermelidir.

265

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*RootFS, initramfs*

BBB



```
# RootFS.initramfs = RootFS.skel.initramfs + _install
$ cd $PROJECT_HOME/RootFS.initramfs
$ rm -fr *
$ cp -a ../RootFS.skel.initramfs/* .
$ cp -a $BUSYBOX_OUT/_install/* .
$ rm -f linuxrc
$ chmod +s bin/busybox || exit 1
$ rm -fr /tftpboot/RootFS.initramfs
$ cp -a RootFS.initramfs /tftpboot
$ echo "Çekirdeği tekrar derle."
```

266

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*RootFS, BR**BBB*

```
# clone
# İlk kuruluř ooooook uzun srer.

$ git clone git://git.buildroot.net/buildroot

# menuconfig, build

$ make -C $BR_SRC O=$BR_OUT clean          # !!!
$ make -C $BR_SRC O=$BR_OUT menuconfig
$ make -C $BR_SRC O=$BR_OUT -j2

# Her bir paket ayrıca derlenebilir.

$ make -C $BR_SRC O=$BR_OUT busybox-menuconfig
```

267

Göml Linux eęitimleri, kuruluřu ve desteęi

<http://ucanlinux.com>*RootFS, BR**BBB*

```
# update RootFS
#

# Temiz bir RootFS dizini kur.
#
$ rm -fr RootFS
$ mkdir RootFS
cd RootFS

# BR'nin rettięi rootfs.tar dosyasını RootFS iine a.
#
$ tar -x -f $PROJECT_HOME/br/output/images/rootfs.tar

# Varsa gncellemeleri yap.
# Gncellemeler el ile veya daha iyisi bir betik
# yardımı ile yapılmalıdır.
#
$ PROJECT_HOME/update.rootfs
```

268

Göml Linux eęitimleri, kuruluřu ve desteęi

<http://ucanlinux.com>

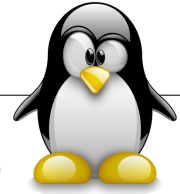
*RootFS, BR**BBB*

```
# Nihai RootFS'i /tftpboot altına taşı.
#
$ rm -fr /tftpboot/RootFS
$ cp -a RootFS /tftpboot
$ ls -l /tftpboot/RootFS
```

- > RootFS hazırdır ve aşağıdaki gibi kullanılabilir. ağ üzerinden NFS ile mount edilebilir. ubi imajı haline getirilip, NAND'a atılabilir. çekirdeğe eklenip initramfs olarak kullanılabilir. ayrık initramfs olarak kullanılabilir. MMC'ye atılıp oradan kullanılabilir.
- ...

269

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*RootFS, update**BBB*

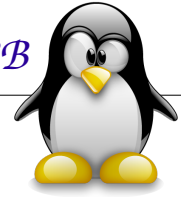
- > /etc/securetty içine seri kanal ismi eklenebilir. ttySAC0, pts/0, pts/1
- > Eğer R0 bir sistem varsa veya history'nin reboot sırasında silinmesi isteniyorsa, /root altındaki .bash* ve .ash* silinir. .ash_history dosyası /tmp'ye yönlendirilir.


```
$ cd /root
$ ln -s -f /tmp/ash_hisyory .ash_history
```
- > /root/.profile dosyası PS1 vs için güncellenebilir.
- > /etc/issue dosyasına açılış mesajı konabilir.
- > /linuxrc gibi gereksiz dosyalar silinebilir.
- > /etc/fstab amaca göre güncellenebilir.

270

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

*RootFS, update**BBB*

- > devfs kullanılacağı için /dev içeriği silinebilir.
- > Uygulamaları başlatan betikler, /etc/init.d/ altına yazılabilir.
- > ...

271

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*MMC, Partitions**BBB*

- > OMAP3x boot rom'un çalışabilmesi için bölümlendirme tablosunda özel bir geometri, 255 head ve 63 sector/track olmalı.
- > sfdisk, cdisk, fdisk, parts vs gibi pek çok program ile MMC'ye MS-DOS bölümlendirmesi yapılabilir.
- > Anlaşılması daha kolay olduğu için fdisk kullanılacaktır.
- > fdisk programına << ile betik ortamlarında, etkileşim olmadan da kullanılabilir.
- > 1. bölümün 32M ve ikinci bölümün 128M olduğunu kabul edelim ve ilk 512 baytı sıfırlayalım.


```
$ BOOT_SIZE=32M
$ ROOT_SIZE=128M
$ dd if=/dev/zero of=$MMC_DEVICE bs=512 count=1
```

272

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

MMC, Make Partitions

BBB



```
$ fdisk -H255 -S63 $MMC_DEVICE << EOF
o
n
p
1

+$BOOT_SIZE
a
1
t
c
x
b
1
63
r
n
p
2

+$ROOT_SIZE
p
w
EOF
```

273

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*MMC, Fill Boot Partition*

BBB



```
$ MMC_DEVICE=/dev/mmcblk0

$ mkfs.vfat -F32 -n "UL.BOOT" ${MMC_DEVICE}p1
# mkfs.ext2 -L "UL.Boot" ${MMC_DEVICE}p1

$ mount ${MMC_DEVICE}p1 $MP

$ cp $UBOOT_OUT/MLO $MP
$ cp $UBOOT_OUT/u-boot.bin $MP
$ cp /tftpboot/uImage $MP/uImage

$ cat <<EOF > /tmp/uEnv.txt
setenv bootargs console=tty00,115200n8
setenv uenvcmd 'fatload mmc 0:1 80007fc0 uImage && bootm 80007fc0'
EOF

$ cp /tmp/uEnv.txt $MP
$ df $MP

$ umount $MP || exit 1
```

274

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

.configs

BBB



> Board başarı ile açıldıktan sonra, aşağıdaki bilgilerin saklanması ilerisi için faydalı olacaktır.

1.kernel, br, busybox ve br/busybox içindeki .config dosyaları saklanmalıdır.

2.dmesg çıkış saklanmalıdır.
Özellikle cmdline bilgisi çok faydalı olacaktır.

3./proc/partitions saklanabilir.

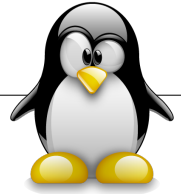
4.u-boot çevre değişkenleri mutlaka saklanmalıdır.

5....

275

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



Hatırlatmalar

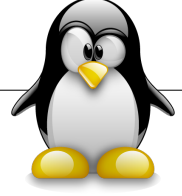
Biz bunları zaten biliyoruz diyenlere...

276

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Terminaller



- Grafik ekrandan kara ekrana geçmek için Ctrl+Alt+F1 ... Ctrl+Alt+F6
- Kara ekranda grafik ekrana geçmek için Alt+F7
- Kara ekranlar arası dolaşmak için Alt+F1..Alt+F6
- Kara ekran sayısını değiştirmek için /etc/inittab'ı güncelle.

277

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Çevre Değişkenleri



- Kabuk içinde bulunan bazı değişkenler bir tabloda saklanabilir.
Bu tabloya çevre değişkenleri tablosu denir.
- Tablonun amacı, kabuk içinde çalışan proseslerin bu değişkenlere erişebilmesini sağlamaktır.

SHELL	/bin/bash
USER	nazim
PWD	/nk/home/nazim
LANG	en_US.UTF-8
HOME	/home/nazim
LOGNAME	nazim
LC_TIME	tr_TR.UTF-8

278

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Çevre Değişkenleri



- › Kabuk değişkenleri (shell variables) genelde ufak harflerle, export'a girecek çevre değişkenleri (env. variables) genelde büyük harflerle verilir.

```
# Listeleme
$ env

# Tabloya değişken ekleme
$ export EGITIM="linux eğitimi"
$ export A=3

# İnceleme
$ echo $EGITIM
$ echo A
$ echo $A
$ env | grep EGITIM
```

279

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Çevre Değişkenleri



```
# Silme
$ export -n EGITIM
$ echo $EGITIM
$ env | grep EGITIM

# Tabloya sonradan değişken ekleme.
$ S1=abc
$ S1="$S1:abc"
$ export $1

$ unset S1
# Hem export tablosundan hem de kabuk içinden siler.
```

280

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Çevre Değişkenleri



> Bazı standard çevre değişkenleri

```
LD_LIBRARY_PATH: Shared library search path.
DISPLAY       : Screen id for X.
EDITOR        : Default editor.
HOME          : Current user home directory.
HOSTNAME      : Name of the local machine.
MANPATH       : Manual page search path.
SHELL         : Current shell name.
TERM          : Current terminal type.
USER          : Current user name.
PATH          : Search path for commands.
```

> C tarafında değişken elde etmek.

```
#include <stdlib.h>
char *getenv(const char *name);
```

281

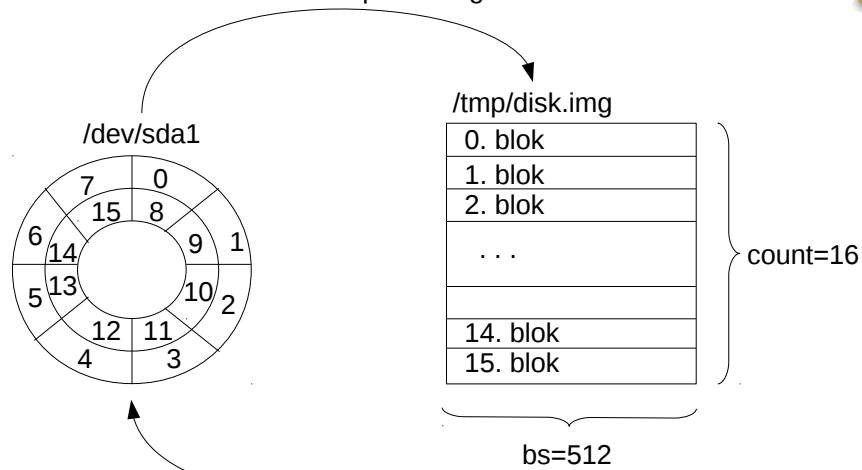
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Device Dump || Device Destroy



```
$ dd if=/dev/sda1 of=/tmp/disk.img count=16 bs=512
```



```
$ dd if=/tmp/disk.img of=/dev/sda1 count=16 bs=512
```

282

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: stdin/out/err files



```
#include <stdio.h>

int main(int argc, char *argv[]){

    FILE *stdin, *stdout, *stderr;

    stdin= fopen("/dev/pts/0", "r");
    stdout= fopen("/dev/pts/1", "w");
    stderr= fopen("/dev/pts/2", "w");

    ...

    on_exit:
    fclose(stdin);
    fclose(stdout);
    fclose(stderr);

}
```

283

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: stdin/out/err files



```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]){

    fprintf(stdout, "stdin fileno: %d\n", fileno(stdin));
    fprintf(stdout, "stdout fileno: %d\n", fileno(stdout));
    fprintf(stdout, "stderr fileno: %d\n", fileno(stderr));

    return EXIT_SUCCESS;

}
```

284

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: stdin/out/err files

```
$ ./fileno
stdin fileno: 0
stdout fileno: 1
stderr fileno: 2
```

```
$ ls -l /proc/$$/fd
total 0
lrwx----- 1 nazim nazim 64 Eyl  8 23:54 0 -> /dev/pts/0
lrwx----- 1 nazim nazim 64 Eyl  8 23:54 1 -> /dev/pts/0
lrwx----- 1 nazim nazim 64 Eyl  8 23:54 2 -> /dev/pts/0
lrwx----- 1 nazim nazim 64 Eyl  9 00:08 255 -> /dev/pts/0
```

285

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: static compiling*

- › Kütüphaneler, derleme sırasında otomatik olarak kodun içine gömülürler.
Kod gömülü sisteme taşınırken, tek başına taşınır.
Herhangi bir kütüphane taşımaya ihtiyaç yoktur.

```
$ gcc -O2 -Wall -o dene dene.c
```

```
$ ls -l dene
-rwxrwxr-x 1 nazim nazim 7450 Eyl  9 21:15 dene
```

```
$ gcc -O2 -Wall -o dene --static dene.c
```

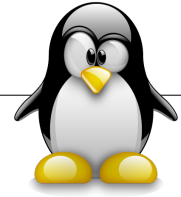
```
$ ls -l dene
-rwxrwxr-x 1 nazim nazim 779629 Eyl  9 21:16 dene
```

286

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: stripping



```
$ file dene
```

```
dene: ELF 32-bit LSB executable,
      Intel 80386,
      version 1 (GNU/Linux),
      statically linked,
      for GNU/Linux 2.6.24,
      BuildID[sha1]=0xcea9356c60acfd86e5a7c1eb0689186d74b1a2b1,
      not stripped
```

```
$ objcopy --strip-all dene
```

```
$ ls -l dene
```

```
-rwxrwxr-x 1 nazim nazim 709232 Eyl  9 21:24 dene
```

```
$ file dene
```

```
stripped özelliğini gör!
```

- Gömülü sistemlere atılan her program mutlaka strip (soyma) işleminden geçirilmelidir ki disk ve bellekte az yer kaplasın.

287

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: kabuk kavramı



- Dosyadan veya standard girişten okuduğu komutları yürüten yorumlayıcıya veya yazılım aracına kabuk denir. Örneğin bash, ksh,...

- Betik başlarında yorumlayıcı isimleri #! ile verilir.

```
#!/bin/bash
#!/bin/perl
```

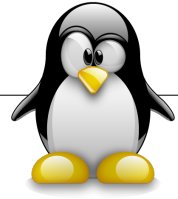
- Temel Yapı:

```
while(1){
  echo PS1
  read Command
  if ( Command == exit ) break;
  if ( & exist ) execute_background(Command);
  else execute_foreground(Command);
}
```

288

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: jobs

- > Her kabuk arka plandaki işleri l'den başlayarak numaralar.
- > Bu numaralar PID numarası gibi kullanılabilir. PID'den ayırd edebilmek için %No kullanılır.

```
$ jobs # + current job, - prev. job
```

```
$ fg
$ fg %n
```

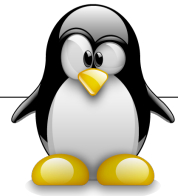
```
$ Ctrl + Z
$ bg
```

```
$ kill -l
$ kill %n
$ kill PID
```

```
$ ps -ef
$ pstree -p
$ top
```

289

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: jobs*

```
$ ps -ux # current user's processes.
```

```
$ ps -ef
$ ps -aux # List all.
```

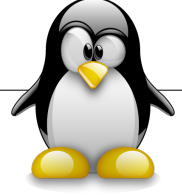
```
$ ps -aux | grep nazim | grep bash
```

```
PID: Process ID
VSD: code+data+stack size
RSS: KB in RAM
TTY: Terminal
R : Runnable
S : Sleep
W : Paging
Z : Zombie
```

290

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: düzenli ifadeler

.	herhangi bir karakter
[...]	içerdeki herhangi bir karakter
[^...]	içerde olmayan herhangi bir karakter, ^:not
^	satır başı
\$	satır sonu
A*	A ifadesi 0 veya daha fazla meydana çıkarsa
A^	1 veya daha fazla
A?	0 veya 1 kez
...	

291

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: düzenli ifadeler*

```
$ ls /tmp/abc*[1234]
$ ls /tmp/abc*[1-4]
$ tail [^r-t]*.[1-4]
$ ps aux | grep -E 20[1-9]{2} # A{N} A, N kere tekrar eder.
```

292

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: bazı komutların kullanımı

```

$ ls -t      # time, en yeni dosya en önce
$ ls -S      # En büyük en önce
$ ls -lrt    # En yeni en dipte, r:reverse order

$ cat <<EOF > /tmp/giris
bu
bir
giriş
denemesidir.
Ctrl+D

$ less file1 file2 # more'dan daha yetenekli.

$ head -n file    # Varsayılan n=10

$ tail -n file
$ tail -f file    # Çok faydalı.

$ wc file # line, word, char. counts. Genelde -l ile
kullanılır.

```

293

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: bazı komutların kullanımı*

```

$ grep pattern file # İçinde pattern geçen satırları listeler.
$ grep -v pattern file # pattern geçmeyen satırları listeler.
$ grep -c pattern file # pattern geçen satırları sayar.

$ sort < file1 > file2
$ sort -r > file2 # reverse sort.
$ sort -ru > file2 # reverse, unique

$ tee file # Çıktılar, aynı anda hem stdout hem de
# dosyaya aktarılır.

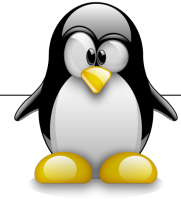
$ time command
real(actual elapsed time)
user(CPU time, program)
sys(CPU time, kernel)
real= user + sys + (i/o wait + running other task)

```

294

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: bazı komutların kullanımı

```

$ which command
$ which falan

$ du -ks * # disk usage

$ df # disk free for the file system
$ df .
$ df -h # h: human readable

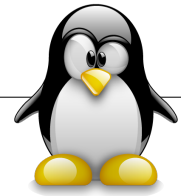
# *.gz
$ gzip file # Dizin sıkıştırılmaz, genelde tar ile kullanılır.
$ gzip -9 file
$ gzip -9 prog -c > prog.gz
$ file prog.gz
$ gunzip prog.gz

# *.bz2
$ bzip2 file # gzip'den %20-25 daha iyi.
$ bunzip2 file

```

295

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: bazı komutların kullanımı*

```

# *.lzma
$ lzma file # bz2'den %10-20 daha iyi.
$ unlzma file

# tape archive
# c: create
# v: verbose
# f: file
# t: test
# x: extract
$ tar cvf test.tar file1 file2 dir1 dir2 dir3 ...
$ tar tvf test.tar
$ tar xvf test.tar

# tar ve compress ortak kullanılabilir.
# z: *.gz
# j: *.bz2
# --lzma: *.lzma
# - : output file is stdout
$ tar zcvf etc.tar.gz /etc
$ tar cvf - /etc | gzip -9 -c > etc.tar.gz

```

296

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: bazı komutların kullanımı

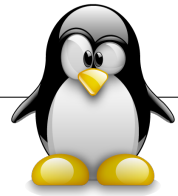
```
# ASCII to PS
# Özellikle bilgisayar programlarını yazıcıdan kullanışlı biçime
# almak için kullanılır.
$ a2ps test1.c test2.c test.h -o test.ps
$ ps2pdf test.ps
$ gv test.ps
$ evince test.pdf
$ xpdf test.pdf
$ pdf2ps test.pdf

# "*.c" ile *.c farklıdır!
$ find . -name "*.c"
$ find . -name "*.c" -exec evince {} \;

# locate, veritabanı kullandığı için çok hızlıdır.
$ locate "*.mp3"
$ locate "*Qt*.so"
```

297

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: bazı komutların kullanımı*

```
$ id
$ id user
$ groups

# logout yapmadan başka kullanıcıya geçmek.
# - Yeni kullanıcının bütün tanımlarını kullan.
#   Diğer bir deyişle, .bash_profile işlet.
$ su user
$ su - user
$ su - root
$ su -          # su - root ile aynıdır.

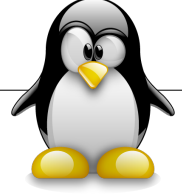
# -c: continue
# -m: mirrors a site
# -r: recursively
# -np: no parent, followlinks in the current directory.
$ wget -c http://slacks.net/slack.iso
```

298

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: bazı komutların kullanımı



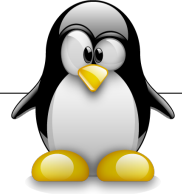
```
# check integrity. 128 bit checksum.  
$ md5sum *.iso > SUM  
$ md5sum -c SUM  
  
# md5sum yerine shaXsum'da kullanılabilir.  
$ sha256sum /etc/passwd
```

299

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: history



- > Ctrl+r ile geçmiş komutlar üzerinde tersten arama yapılabilir.
- > Bash kabuğunda geçmiş komutlar `.bash_history` dosyasına saklanır.
- > C programlama içinden readline kütüphanesi ile history desteği verilebilir.

```
$ history # bütün geçmiş komutlar.
```

```
$ !! # son komutu işlet.
```

```
$ !5 # 5 nolu komutu işlet.
```

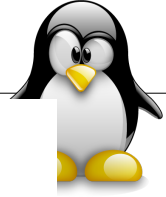
```
# !cp # cp ile başlayan son komutu işlet.
```

300

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: vi



vi basic commands

Summary of most useful commands

©Copyright 2006-2005, Free Electrons, <http://free-electrons.com>. Latest update: Sep 15, 2009
Free to share under the terms of the Creative Commons Attribution-ShareAlike 2.5 license
(<http://creativecommons.org/licenses/by-sa/2.5/>)
Sources, translations and updates on our free training materials: http://free-electrons.com/training/tutro_ums_linux
Thanks to: Inho Chen.

Entering command mode

[Esc] Exit editing mode. Keyboard keys now interpreted as commands.

Moving the cursor

h (or left arrow key) move the cursor left.
l (or right arrow key) move the cursor right.
j (or down arrow key) move the cursor down.
k (or up arrow key) move the cursor up.
[Ctrl] f move the cursor one page forward.
[Ctrl] b move the cursor one page backward.
^ move cursor to the first non-white character in the current line.
\$ move the cursor to the end of the current line.
G go to the last line in the file.
nG go to line number *n*.
[Ctrl] G display the name of the current file and the cursor position in it.

Entering editing mode

i insert new text before the cursor.
a append new text after the cursor.
o start to edit a new line after the current one.
O start to edit a new line before the current one.

Replacing characters, lines and words

r replace the current character (does not enter edit mode).
s enter edit mode and substitute the current character by several ones.
cw enter edit mode and change the word after the cursor.
C enter edit mode and change the rest of the line after the cursor.

Copying and pasting

yy copy (yank) the current line to the copy/paste buffer.
p paste the copy/paste buffer after the current line.
P Paste the copy/paste buffer before the current line.

Deleting characters, words and lines

All deleted characters, words and lines are copied to the copy/paste buffer.

x delete the character at the cursor location.
dw delete the current word.
D delete the remainder of the line after the cursor.
dd delete the current line.

Repeating commands

. repeat the last insertion, replacement or delete command.

Looking for strings

/string find the first occurrence of *string* after the cursor.
?string find the first occurrence of *string* before the cursor.
n find the next occurrence in the last search.

Replacing strings

Can also be done manually, searching and replacing once, and then using *n* (next occurrence) and *.* (repeat last edit).

n,*ps*/*str1*/*str2*/*g* between line numbers *n* and *p*, substitute all (*g*: global) occurrences of *str1* by *str2*.
1,*\$\$*/*str1*/*str2*/*g* in the whole file (*\$\$*: last line), substitute all occurrences of *str1* by *str2*.

Applying a command several times - Examples

5j move the cursor 5 lines down.
30dd delete 30 lines.
4cw change 4 words from the cursor.
1G go to the first line in the file.

Misc

[Ctrl] l redraw the screen.
J join the current line with the next one

Exiting and saving

ZZ save current file and exit vi.
:w write (save) to the current file.
:w *file* write (save) to the *file* file.
:q! quit vi without saving changes.

Going further

vi has much more flexibility and many more commands for power users! It can make you extremely productive in editing and creating text. Learn more by taking the quick tutorial: just type [vimtutor](http://vimtutor.vim.org). Find many more resources on the net!



301

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: alias



- Sık kullanılan komutlar için kısayol tanımlamasıdır.
- Kalıcı olması için bir dosyada veya ~/.bashrc'de tanımlanabilir.
- Bir dosyada tanımlanırsa, dosya aşağıdaki gibi çalıştırılmalıdır.

```
$ . ./test_alias
```

```
$ alias
```

```
$ alias ls="ls -la"
```

```
$ alias cp="cp -i"
```

```
$ alias c="gcc -O2 -Wall -o test.c test"
```

```
$ alias c
```

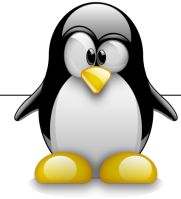
```
$ unalias c
```

```
$ alias c
```

302

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: date

- O anki zamanı döndürür.
- Locale değişkenine göre tarih/zaman bilgisini yazar.

```
$ date
```

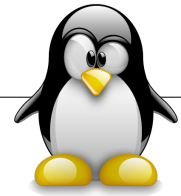
```
# Şu anki zamanı Unix Epoch Time'dan itibaren saniye  
# cinsinden göster.  
# Unix Epoch Time : 1 Ocak 1970, gece yarısı  
$ date +%s
```

```
# Dosya ismi üretmek için kullanılabilir.
```

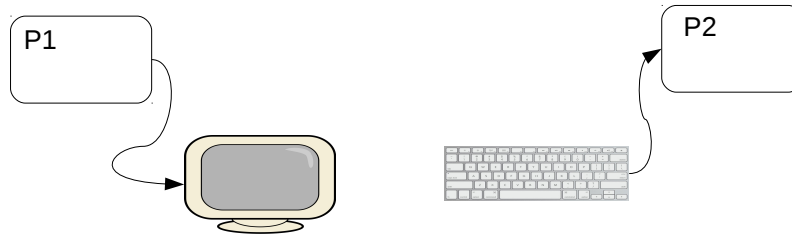
```
F="backup.`date +%F-%H-%M-%S`.tgz"  
tar zcvf /mnt/usb/$F workspace/project
```

303

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: pipes (unnamed)*

- Bir prosesin stdout çıkışını başka bir programın stdin çıkışına bağlama işi pipe ile yapılır.



```
printf(stdout, "merhaba");
```

```
char mesaj[128];  
fscanf(stdin, "%s", mesaj);
```

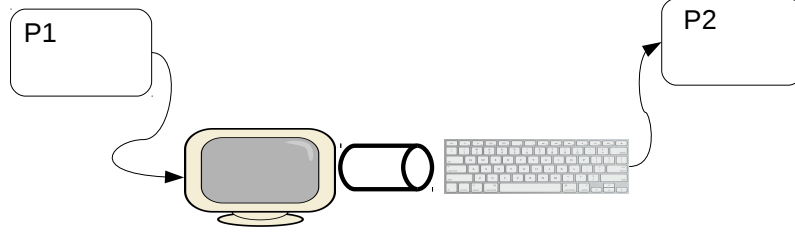
304

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: pipes (unamed)

```
$ P1 | P2
```



```
printf(stdout, "merhaba"); char mesaj[128];
                          fscanf(stdin, "%s", mesaj);
```

305

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: pipes (unamed)*

```
$ cat /etc/passwd | wc -l
```

```
$ ifconfig eth0 | grep HW
eth0      Link encap:Ethernet  HWaddr 00:90:f5:dc:51:d2
```

```
$ cat /etc/passwd | while read X
> do
> echo okunan satır $X
> done
```

```
okunan satır root:x:0:0:root:/root:/bin/bash
okunan satır daemon:x:1:1:daemon:/usr/sbin:/bin/sh
okunan satır bin:x:2:2:bin:/bin:/bin/sh
okunan satır sys:x:3:3:sys:/dev:/bin/sh
....
```

```
$ yes | command
$ yes no | command
$ yes "" | command # Boş olarak enter'a basmak gibidir.
```

306

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: pipes (unnamed)



- Linux komutları prensip olarak tek bir işi çok iyi yapmak üzere programlanmıştır.
- Bu komutlar i/o redirection ile etkileşimli bir biçimde kullanılarak, yani giriş ve çıkışlar, <, >, >>, | gibi ifadelerle yönlendirilerek son derece karmaşık işler yapılabilir.

307

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: symbolic link



- Başka bir dosya veya dizini gösteren özel bir dosyadır.
- Gömülü sistemlerde kullanılan busybox sistemi sembolik linklerin genişçe kullanımına çok güzel bir örnektir.

```
$ ln -s mevcut_dosya sembolik_isim  
$ ln -s mevcut_dosya
```

```
$ rm sembolik_isim # Sadece sembolik ismi siler.
```

- -s kullanılmazsa hard link yaratılır. Hard link tavsiye edilmez.

308

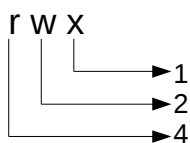
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: mod ve sahiplik



$\underbrace{rwx\ rw-\ r--}_{\text{chmod ile değiştirilir}} \quad \underbrace{\text{can\ user}}_{\text{chown ile değiştirilir}} \quad \dots\ /home/can/dene.c$



owner	group	other
rwx	rw-	r--
$\underbrace{\hspace{2em}}$	$\underbrace{\hspace{2em}}$	$\underbrace{\hspace{2em}}$
4+2+1	4+2+0	4+0+0
7	6	4

309

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: root



- root kullanıcısı her işi yapmaya yetkilidir. UID değeri her zaman 0'dır. Ya da 0 UID değerine sahip bütün kullanıcılar root yetksine sahiptir.
- Gömülü sistemlerde genelde root ile çalışılırken, standard dağıtımlarda root kullanılmaz. Bunun yerine sudo kavramı getirilmiştir.
- root şifresini vermeden bir kullanıcıya root yetkisi verilebilir. Bunun için ilgili kullanıcı /etc/sudoers dosyasına kaydedilir. Böylece bu kullanıcı root şifresini bilmeden sudo komutu ile root yetkisinde iş yapabilir. GS'de genelde sudo kullanılmaz.

```
$ sudo mount /dev/sda3 /mnt/disk3
```

- Standard bir kullanıcı, su - komutu ile root kullanıcısına logout olmadan geçebilir. Sudo yetkili kullanıcı ise sudo -s ile root'a geçebilir.

310

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: login kavramı

login: can

```

rw- r-- --- can user ... /home/can/dene.c

```

Diagram illustrating the login process. The text "login: can" is shown above the permissions "rw- r-- ---" in the file listing. An arrow points from "can" in the login prompt to the "can" user in the permissions. Another arrow points from the "can" user in the permissions to the "can" user in the file listing. The file listing also shows "user" and "... /home/can/dene.c".

311

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: login*

login: oguzhan

/etc/group

user: ali, oguzhan, can

```

rw- r-- --- can user ... /home/can/dene.c

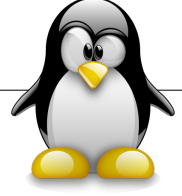
```

Diagram illustrating the login process. The text "login: oguzhan" is shown above the permissions "rw- r-- ---" in the file listing. An arrow points from "oguzhan" in the login prompt to the "oguzhan" user in the /etc/group file listing. Another arrow points from the "oguzhan" user in the /etc/group file listing to the "can" user in the permissions. The /etc/group file listing also shows "user: ali, oguzhan, can". The file listing also shows "can", "user", and "... /home/can/dene.c".

312

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: login

```
login: deniz
      |
      |
      v
rw- r-- --- can user ... /home/can/dene.c
```

313

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: kılavuz sayfaları*

```
$ man N command
$ man -k keyword
$ man -f word
$ echo $MANPATH
```

1. Genel komutlar. pwd, ls, man
2. Sistem çağrıları(çekirdek fonksiyonları) open, read
3. Kütüphane fonksiyonları. fprintf, fopen
4. Özel dosyalar. zero, null, full
5. Dosya formatları. passwd, exports
6. Oyunlar ve ekran koruyucuları. noof, molecule
7. Diğerleri. fifo, iso_8859-9, environ
8. Sistem yönetim komutları ve servisleri. nfsd, mount, mke2fs

> Dosyalar MANPATH ile gösterilen yerde aranılırlar.

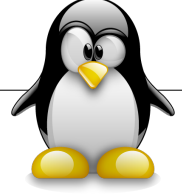
> Çok fazla kılavuz sayfası vardır. Az yer kaplasın diye gzip ile sıkıştırılmışlardır.

314

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: geçici dosyalar



```
$ mktmp
$ mktmp test.XXXXXX
$ ls -l test.*
```

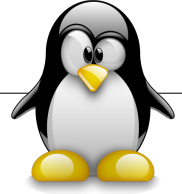
- -d ile geçici dizin de yaratılabilir.
- Dosya ile işlem bitince rm ile dosya silinmelidir.
- C içinde yaratılan geçici dosyalar, proses bitince otomatik olarak silinirler.
- Geçici dosyalar mevcut kullanıcı için rw- modunda yaratılırlar.

315

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: proses kavramı



- Bir programın çalışan örneğine proses denir.
- Linux'de herşey ya dosya ya da procestir.
- Her prosesin biricik pid ve ppid numarası vardır.
- init prosesinin pid numarası her zaman 1'dir ve çekirdek tarafından başlatılır.

```
$ pstree -p
```

- Genelde anne ölünce çocuk da ölür. nohup komutu ile çocuk ölümleri engellenir.

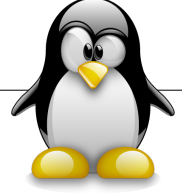
```
$ nohup prog 1>out 2>err &
```

316

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: proses kavramı



- Kabuktan proses başlatma:

```
$ ls  
$ sleep 20 &
```

- Arkada bulunan (çalışan veya geçici olarak durdurulmuş olan) bütün prosesler için kabuk tarafından özel bir numara verilir. Bu numara sadece ve sadece mevcut kabuk için geçerlidir. Bu numaraya "iş numarası" denir. pid yerine kullanılabilir.
- Her proses exit(N) ile son bulur. N değeri 0..255 arasında değişir. N değeri anne procese döner. Hata yoksa genelde 0 kullanılır.

317

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: proses kavramı



- Proses son bulduğunda, bütün açık dosyalar otomatik olarak kapatılır.
- Eğer prosesin çocukları varsa, genelde 1 nolu procese evlatlık verilir.
- Parent procese CHLD sinyali gönderilir.
- Eğer önce anne ölür, sonra çocuk ölürse, çocuk exit değerini ve sinyalini gönderecek bir anne proses bulamaz. Bu tür proseslere zombi denir ve belirli bir süre sonra proses tablosundan otomatik olarak silinirler.

318

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: /proc

- Gerçek, fiziksel bir cihazla ilgili olmayan bir dosya sistemidir. Bundan dolayı /proc veya benzer dosya sistemlerine, pseudo veya "sanki" dosya sistemi denir.
- Bu dosya sistemi tamamen sanal olup, çekirdek tarafından, ya açılış ya da kullanım anında, bellekte kurulur.
- Bundan dolayı pek çok dosyanın yaratılma zamanı, hemen komutun girildiği zamandır.
- Bu tür dosya sistemlerinin esas amacı, kernel space bilgilerine, user space tarafından kolayca erişim sağlamaktır. Ya da her iki alan arasında bilgi alışverişini sağlamaktır.

319

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: /proc*

- Pusedo dosya sistemlerinin içeriğinde bir standardlaşma yoktur. Bir çekirdek sürümünde olan bir dizin veya dosya başka bir sürümde hiç olmayabilir ya da farklı bir biçimde bulunabilir.
- \$ man 5 proc kılavuz sayfası içeriği genişçe açıklar.
- /proc altındaki dosyalar C tarafından, sıradan bir dosya gibi fopen() ile açılarak içeriği kullanılabilir.

```
FILE *f;
f= fopen("/proc/cpuinfo", "r");
read & parse lines...
fclose(f);
```

320

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: /proc

- > Kabuk tarafında ise genelde cat komutu ile dosyalar okunur ve echo komutu ile güncellenir.
- > /proc dosya sisteminde her proses için bir adet dizin bulunur. Bu dizinin ismi pid sayısı ile aynıdır.
- > Bu dosya sistemi genelde prosesler için yaratıldığından, dosya sistemi /proc dizinine mount edilir.
- > \$\$ ifadesi çalışan kabuğun pid numarasını verir.
- > filesystems: Çekirdekte o anda desteklenen dosya sistemlerini listeler. Modül olarak derlenmiş ama henüz yüklenmemiş destekler gösterilmez. Bunun için /proc/config.gz kullanılabilir.

321

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: /proc*

- > uptime: Ayakta kalınan ve boş geçen süreler raporlanır.
- > mounts: Bağlı dosya sistemlerine ait bilgiler.
- > self: 0 anda çalışan prosesin pid değeri. \$\$ ile karıştırma.
- > tty/driver: Seri sürücülere ait bilgiler.
- > sys/kernel, hostname, version, ...

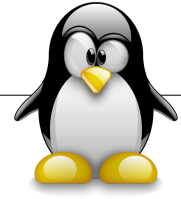
pid/ altında bulunan bazı önemli dosyalar:

- cwd: 0 anda çalışılan dizin.
- environ: Çevre değişkenleri.
- exe: argv[0] bilgisi.

322

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: /proc

- fd: Aktif fd'leri gösterir.

```
$ cd /proc/$$/fd
$ ls -l
$ ls -l /proc/1/fd
$ cd /proc/`pidof udevd`/fd
$ ls -l
```

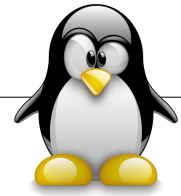
- /proc dizinini kullanan prosesler lsof komutu ile tespit edilebilir.

```
$ lsof /proc
```

- Gömülü Sistemlerde /proc dosya sistemi genelde /etc/rcS gibi bir betik tarafından açıkça mount edilir. Ya da /etc/fstab dosyasına giriş yapılarak mount edilir.

323

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: /proc*

- /proc dosya sisteminin açıkça mount edilmesi.

```
$ mount -t proc proc /proc # veya
$ mount -t proc none /proc
$ mount -t ext2 /dev/sda1 /disk # karşılaştırma için...
```

- /etc/fstab satırı.

```
proc /proc proc defaults 0 0
```

- Birinci 0 değeri dump içindir, kullanılmaz. İkinci 0 değeri fsck içindir.

```
0: fsck yapma.
1: rootfs'tir. fsck yap.
2: En sona bu dosyayı fsck yap.
```

324

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: makefile

- Hedef dosyanın, bağımlılıklara göre nasıl elde edileceğini tarif eden dosyaya Makefile denir.
- > İki temel kavramı vardır: girişler ve bağımlılıklar.

```
# Makefile, girişler
all: derle goster sil

derle:
    echo "programlar derlendi."

goster:
    echo "programları göster."

sil:
    echo "hepsi silindi."
```

325

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: makefile*

```
# Makefile, bağımlılıklar.
# Değişkenler: $(NAME)
# Default target: all

CFLAGS = -g -Wall -Os
PROG = test

all: $(PROG)

# $@: test, target name
# $^: file1.o file2.o, bağımlılıklar
#
# gcc -o test file1.o file2.o -g -Wall -Os
test: file1.o file2.o
    gcc -o $@ $^ $(CFLAGS)

clean:
    $(RM) -f $(PROG)

install:
    cp $(PROG) /usr/local/bin
```

326

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: makefile



```
$ cat file1.c

#include <stdio.h>
#include "project.h"
int main(){
    fprintf(stdout, "program başarılı bitti.\n");
    return 0;
}

$ cat file2.c

#include <stdio.h>
void test(){
    fprintf(stdout, "test tamamlandı.\n");
}

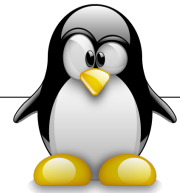
$ cat project.h
void test();
```

327

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: makefile



```
# compile
$ gcc -g -O3 -Wall file1.c file2.c -o project

# install
$ sudo cp project /usr/local/bin

# clean
$ rm project *.o

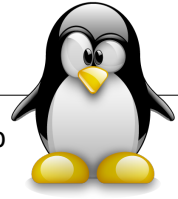
# run
$ cd
$ project
program başarılı bitti.
```

- > Büyük projelerde, bütün bu adımları el ile yapmak çok zor veya çok zahmetli olacaktır.
- > Ayrıca, bir sonraki derlemede, sadece değişiklikten etkilenen dosyalar derlenmelidir. İşte bu iş için Makefile sistemi kullanılır.

328

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: makefile

```
# Bu işleri yapan kaba bir Makefile aşağıdaki gibi yazılıp
# işletilebilir.

# Makefile.00
project: file1.o file2.o
    gcc -Os -g -Wall file1.o file2.o -o project

file1.o: project.h file1.c file2.c
    gcc -c file1.c -o file1.o

file2.o: project.h file2.c
    gcc -c file2.c -o file2.o

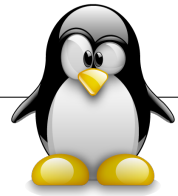
install:
    sudo cp -f project /usr/local/bin

uninstall:
    sudo rm -f /usr/local/bin/project

clean:
    rm -f project *.o
```

329

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: makefile*

```
$ which project

$ make project
gcc -c file1.c -o file1.o
gcc -c file2.c -o file2.o
gcc -Os -g -Wall file1.o file2.o -o project

$ make install
sudo cp -f project /usr/local/bin

$ ls -l project
-rwxrwxr-x 1 nazim nazim 7377 Eyl 14 22:52 project

$ make clean
rm -f project *.o

$ ls -l project
ls: cannot access project: No such file or directory
```

330

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: makefile

```
$ which project
/usr/local/bin/project

$ project
program başarılı bitti.

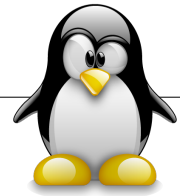
$ make uninstall
sudo rm -f /usr/local/bin/project

$ which project

$ project
bash: /usr/local/bin/project: No such file or directory
```

331

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: makefile*

```
# Makefile.01

PROG = project
CFLAGS= -Os -g -Wall
LDFLAGS = -lm
RM = rm -f
CP = cp -f
CC = gcc

all: $(PROG) install

$(PROG): file1.o file2.o
    $(CC) $(LDFLAGS) $(CFLAGS) $^ -o $(PROG)

file1.o: file1.c project.h file2.c
    $(CC) $(CFLAGS) -c $< -o $@

file2.o: file2.c project.h
    $(CC) $(CFLAGS) -c $< -o $@
```

332

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: makefile

```
# Makefile.01

install:
    sudo $(CP) $(PROG) /usr/local/bin

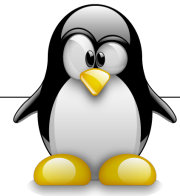
uninstall:
    sudo $(RM) /usr/local/bin/$(PROG)

clean:
    $(RM) $(PROG) *.o

$ make
gcc -Os -g -Wall -c file1.c -o file1.o
gcc -Os -g -Wall -c file2.c -o file2.o
gcc -lm -Os -g -Wall file1.o file2.o -o project
sudo cp -f project /usr/local/bin
```

333

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: makefile*

- > Son verilen Makefile dosyası genel olmasına rağmen sadece native kod üretir.
- > Çapraz derleme için CROSS_COMPILE değişkeni Makefile içine eklenebilir.
- > Cross derleme için son Makefile içinde CC değişkenini aşağıdaki gibi değiştirmek yeterlidir.

```
CC = $(CROSS_COMPILE)gcc
```

- > Bu durumda aşağıdaki gibi hem native hem de cross derleme yapılabilir.

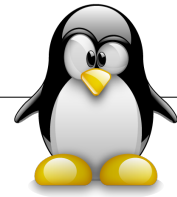
```
$ make # native derler.
$ file project
```

```
$ make CROSS_COMPILE=arm-linux-gnueabi- # cross derler.
$ file project
```

334

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: makefile

```

$ make
gcc -Os -g -Wall -c file1.c -o file1.o
gcc -Os -g -Wall -c file2.c -o file2.o
gcc -lm -Os -g -Wall file1.o file2.o -o project
sudo cp -f project /usr/local/bin

$ file project
project: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked (uses shared libs),
for GNU/Linux 2.6.24,
BuildID[sha1]=0xdfdc2158739556f903b8735284e5d2ec561dcc7a,
not stripped

$ make clean
rm -f project *.o

$ make CROSS_COMPILE=arm-linux-gnueabi-
arm-linux-gnueabi-gcc -Os -g -Wall -c file1.c -o file1.o
arm-linux-gnueabi-gcc -Os -g -Wall -c file2.c -o file2.o
arm-linux-gnueabi-gcc -lm -Os -g -Wall file1.o file2.o -o project
sudo cp -f project /usr/local/bin

```

335

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: makefile*

```

$ file project

project: ELF 32-bit LSB executable,
ARM, version 1 (SYSV),
dynamically linked (uses shared libs),
for GNU/Linux 2.6.32,
BuildID[sha1]=0x3f611dee63bb4b3489118e328c2c626d200f1467,
not stripped

# Sadece güncellemeden etkilenen dosyalar derlenir.

$ make clean
$ vi file1.c
$ make
gcc -Os -g -Wall -c file1.c -o file1.o
gcc -lm -Os -g -Wall file1.o file2.o -o project
sudo cp -f project /usr/local/bin

```

336

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: makefile

- Büyük projeler için el yordamı ile Makefile kurmak çok zahmetlidir.
- Büyük Makefile dosyaları genelde ide veya otomatik Makefile araçları tarafından kurulur. Autotools, CMake vs.
- Bu tür programlarla kurulan paketler genelde aşağıdaki gibi derlenip kurulabilirler. İlk 3 satır mutlaka mevcuttur.

```

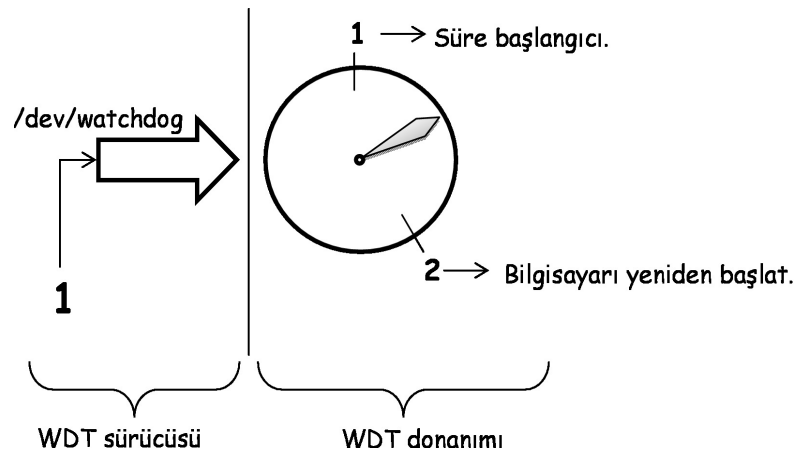
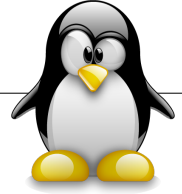
$ ./configure
$ make
$ sudo make install

$ make test
$ make clean
$ make mrproper
$ sudo make uninstall

```

337

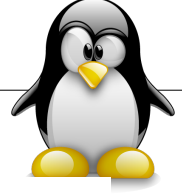
Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>*Hatırlatma: wdt*

338

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Hatırlatma: wdt

```
f=open(/dev/watchdog);
```

} WDT, 1 konumunda işlemeye başlar.

```
A:
...
if (...) goto B;
...
...
write(f, '1');
goto A;
```

} Süre-aşımı (margin, timeout) bölgesi.
WDT sayacı, 2'ye varmadan evvel, write(f, 1) ile, sayaç tekrar 1 konumuna getirilir.
goto B; ile süre-aşımı bölgesinden çıkılır.

```
B:
write(f, 'v');
close(f);
```

} WDT'yi kapatmadan evvel v harfi gönderilir. close() ile WDT kapatılır.

339

Gömülü Linux eğitimleri, kuruluşu ve desteği

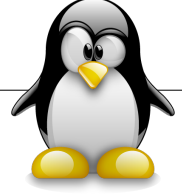
<http://ucanlinux.com>*login gelmiyor?*

- > Çekirdek parametresi olan console tanımsız veya uygun değil
- > /etc/securetty'de tty tanımı yok.
- > busybox modu x değil veya root yetkisinde değil veya kütüphaneler eksik veya kütüphaneler uyumsuz.
- > init= parametresinin gösterdiği program mevcut değil, x modunda değil, ...
- > /init yok, /sbin/init yok, /linuxrc yok, ...
- > inittab'da seri konsol için getty tanımı yok.
- > ...

340

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>



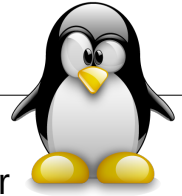
Kapanış

341

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Sorular



Bütün eğitim notları ve gömülü Linux ile ilgili diğer belgeler
<http://ucanlinux.com>
adresinden temin edilebilir.

Her türlü bilgi talebi, düzeltme, eleştiri ve soru

nazim.koc@gmail.com

veya

nazim@ucanlinux.com

adreslerine gönderilebilir.

342

Gömülü Linux eğitimleri, kuruluşu ve desteği

<http://ucanlinux.com>

Telif Hakları



› Bu slaytlarda bulunan bütün içeriğin telif hakları Nazım KOÇ'a aittir.

› Slaytların tamamı veya bir kısmı,

ucanlinux.com adresi kaynak gösterilerek,
üzerinde değişiklik yapılmadan,

ticari veya ticari olmayan her ortamda,
herhangi bir izne gerek kalmadan,
kullanılabilir,
dağıtılabilir.

file: /nk/workspace/notebook/linux.egitimi/gomulu.linux.egitimi/belge.odp
date: 05.07.14

343