

# Linux i386 Önyükeme Kodu NASIL

Yazan:  
**Feiyun Wang**  
<feiyunw (at) yahoo.com>

Çeviren:  
**A.Kemal Ökmen**  
<kemal (at) comu.edu.tr>

Mart 2006

## Özet

Bu belge Linux i386 önyükeme kodu için bir çalışma kılavuzu ve kaynak kod açıklamaları sunar. C-benzeri yalancı kaynak kod açıklamalarına ek olarak, çekirdek geliştirme ile ilgili araç silsilelerinin önemli notlarını ve belirtilmelerini de sunar.

Aşağıdaki konularda yardımcı olmak için tasarlanmıştır:

- Çekirdek konusunda acemilerin Linux i386 önyükeme kodunu anlaması ve
- Çekirdek konusunda tecrübelilerin Linux önyükeme işlemini hatırlamaları.

## Konu Başlıkları

<b>1. Giriş</b>	4
1.1. Teşekkür	4
1.2. Geribildirim	4
1.3. Çeviriler	4
<b>2. Linux Makefile'lar</b>	4
2.1. linux/Makefile	4
2.2. linux/arch/i386/vmlinux.lds	6
2.3. linux/arch/i386/Makefile	7
2.4. linux/arch/i386/boot/Makefile	8
2.5. linux/arch/i386/boot/compressed/Makefile	9
2.6. linux/arch/i386/tools/build.c	10
2.7. Kaynakça	11
<b>3. linux/arch/i386/boot/bootsect.S</b>	11
3.1. Bootsect'i Taşı	11
3.2. Disk parametrelerini Al	12
3.3. Kurulum Kodunu Yükle	13
3.4. Sıkıştırılmış İmgeyi Yükle	13
3.5. Ayarlara Git	14
3.6. Diski Oku	14
3.7. Bootsect Yardımcısı	16
3.8. Muhtelif	17
3.9. Kaynakça	18
<b>4. linux/arch/i386/boot/setup.S</b>	18

4.1. Başlık	18
4.2. Kod Bütünlüğünü Kontrol Et	21
4.3. Yükleyici Türünü Kontrol Et	22
4.4. Bellek Boyutunu	23
4.5. Donanım Desteği	24
4.6. APM Desteği	25
4.7. Korumalı Kip için Hazırlık	26
4.8. A20'yi Etkinleştir	27
4.9. Korumalı Kipe Geç	28
4.10. Muhtelif	29
4.11. Kaynakça	32
<b>5. linux/arch/i386/boot/compressed/head.S</b>	32
5.1. Sıkıştırılmış Çekirdeğin Açılması	32
5.2. gunzip()	35
5.3. inflate()	36
5.4. Kaynakça	38
<b>6. linux/arch/i386/kernel/head.S</b>	39
6.1. Sayfalamayı Etkinleştir	39
6.2. Çekirdek Parametrelerini Al	41
6.3. İşlemci Türünü Kontrol Et	42
6.4. Çekirdeği Başlat	43
6.5. Muhtelif	44
6.6. Kaynakça	46
<b>7. linux/init/main.c</b>	46
7.1. start_kernel()	46
7.2. init()	49
7.3. cpu_idle()	50
7.4. Kaynakça	51
<b>8. SMP Önyükeme</b>	51
8.1. smp_init() Öncesi	51
8.2. smp_init()	53
8.3. linux/arch/i386/kernel/trampoline.S	56
8.4. initialize_secondary()	57
8.5. start_secondary()	58
8.6. Kaynakça	58
<b>A. Çekirdek Derleme ve Kurulum Örneği</b>	59
<b>B. Dahili İlintileme Betiği</b>	60
<b>C. GRUB ve LILO</b>	64
C.1. GNU GRUB	64
C.2. LILO	65
C.3. Kaynakça	66
<b>D. SSS</b>	66
GNU Free Documentation License	66

**Geçmiş**

1.0	Mart 2006	akö
İlk çeviri		

**Geçmiş**

1.0	2004-02-19	FW
İlk yayın, LDP tarafından gözden geçirildi		
0.3.3	2004-01-23	fyw
<code>decompress_kernel()</code> ayrıntıları eklendi; TLDP son gözden geçirme hataları düzeltildi.		
0.3	2003-12-07	fyw
SMP, GRUB ve LILO içerikleri eklendi; Düzeltme ve geliştirme.		
0.2	2003-08-17	fyw
Linux 2.4.20'ya uyarlandı.		
0.1	2003-04-20	fyw
DocBook XML biçimine dönüştürüldü.		

**Yasal Açıklamalar**

Bu belgenin, *Linux i386 Önyükeme NASIL* çevirisinin 1.0 sürümünün **tefif hakkı © 2006 A. Kemal Ök-men'e**, özgün İngilizce sürümünün **tefif hakkı © 2003, 2004 Feiyun Wang'a** aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Özgür Belgeleme Lisansının 1.2 ya da daha sonraki sürümünün koşullarına bağılı kalarak kopyalayabilir, dağıtabilir ve/veya değıştirebilirsiniz. Bu Lisansın bir kopyasını [GNU Free Documentation License](#) (sayfa: 66) başılıklı bölümde bulabilirsiniz.

BU BELGE "ÜCRETSİZ" OLARAK RUHSATLANDIĞI İÇİN, İÇERDİĞİ BİLGİLER İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGEYİ "OLDUĞU GİBİ", AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BİLGİNİN KALİTESİ İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATALI BİLGİDEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

İLGİLİ KANUNUN İCBAR ETTİĞİ DURUMLAR VEYA YAZILI ANLAŞMA HARİCİNDE HERHANGİ BİR ŞEKİLDE TELİF HAKKI SAHİBİ VEYA YUKARIDA İZİN VERİLDİĞİ ŞEKİLDE BELGEYİ DEĞİŞTİREN VEYA YENİDEN DAĞITAN HERHANGİ BİR KİŞİ, BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİLERİN BAŞKA BİLGİLERLE UYUMSUZ OLMASI) YÜZÜNDEN OLUŞAN GENEL, ÖZEL, DOĞRUDAN YA DA DOLAYLI HERHANGİ BİR ZARARDAN, BÖYLE BİR TAZMİNAT TALEBİ TELİF HAKKI SAHİBİ VEYA İLGİLİ KİŞİYE BİLDİRİLMİŞ OLSA DAHİ, SORUMLU DEĞİLDİR.

Tüm telif hakları aksi özellikle belirtilmediğı sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiğı anlamında görülmemelidir.

## 1. Giriş

Bu belge Linux i386 önyükeme kodu hakkında bir çalışma kılavuzu ve kaynak kod açıklamaları sunar. C—benzeri yalancı kaynak kod açıklamalarına ek olarak, çekirdek geliştirme ile ilgili araç silsilelerinin önemli notlarını ve belirtimlerini de sunar. Aşağıdaki konularda yardımcı olmak için tasarlanmıştır:

- Çekirdek konusunda acemilerin Linux i386 önyükeme kodunu anlaması ve
- Çekirdek konusunda tecrübelilerin Linux önyükeme işlemini hatırlamaları.

Bu belgede Linux 2.4.20 temel almıştır.

Bu belgenin proje sanalyöresi [China Linux Forum](#)<sup>(B3)</sup> tarafından sunulmaktadır. Belge ile ilgili çalışmalar ayrıca yazarın kişisel sanalyöresi [Yahoo! GeoCities](#)<sup>(B4)</sup>'de bulunabilir.

### 1.1. Teşekkür

Kılavuzun çevirisi sırasında yardımlarını esirgemeyen [Necdet Yücel](#)<sup>(B5)</sup>'e teşekkür ederim.

### 1.2. Geribildirim

Bu belge için geribildirim memnuniyetle karşılanır. Eklèmelerinizi, yorumlarınızı ve eleştirilerinizi aşağıdaki eposta adresine gönderiniz:

- Feiyan Wang <[feiyunw \(at\) yahoo.com](mailto:feiyunw@yahoo.com)>

### 1.3. Çeviriler

Özgün İngilizce belge ve Türkçe çevirisi.

## 2. Linux Makefile'lar

Linux kodunu incelemeden önce Linux'un nasıl oluşturulduğu, derlendiği ve bağlandığı konusunda bazı temel fikirlere sahip olmalıyız. Bunu en baştan başlayarak yapmanın bir yolu Linux makefile'ları anlamaktır. Çevrimiçi kaynak gösterimini tercih ediyorsanız [Cross-Referencing Linux](#)<sup>(B6)</sup> bağlantısını kontrol edin.

### 2.1. linux/Makefile

Bu üst seviye için bazı iyi bilinen makefile hedefleri şunlardır:

`xconfig, menuconfig, config, oldconfig`

`linux/.config` çekirdek yapılandırma dosyasını üretir;

`depend, dep`

`alt` dizinlerde `linux/.depend`, `linux/.hdepend` ve `.depend` gibi bağımlılık dosyalarını üretir;

`vmlinux`

en önemli hedef olan `linux/vmlinux` yerleşik çekirdek görüntüsünü üretir;

`modules, modules_install`

`/lib/modules/$ (KERNELRELEASE)` içerisindeki modülleri üretir ve kurar;

`tags`

[vim](#)<sup>(B7)</sup> ile kaynağın görüntülenmesi için `linux/tags` etiket dosyasını üretir.

linux/Makefile anahatları ile aşağıdaki gibidir:

```
include .depend
include .config
include arch/i386/Makefile

vmlinux: linux/vmlinux üretir
/* "stext" giriş noktası arch/i386/kernel/head.S dosyasında tanımlı*/
$(LD) -T $(TOPDIR)/arch/i386/vmlinux.lds -e stext
/* $(HEAD) */
+ from arch/i386/Makefile
    arch/i386/kernel/head.o
    arch/i386/kernel/init_task.o
init/main.o
init/version.o
init/do_mounts.o
--start-group
/* $(CORE_FILES) */
+ from arch/i386/Makefile
    arch/i386/kernel/kernel.o
    arch/i386/mm/mm.o
kernel/kernel.o
mm/mm.o
fs/fs.o
ipc/ipc.o
/* $(DRIVERS) */
drivers/...
    char/char.o
    block/block.o
    misc/misc.o
    net/net.o
    media/media.o
    cdrom/driver.o
    and other static linked drivers
+ from arch/i386/Makefile
    arch/i386/math-emu/math.o (ifdef CONFIG_MATH_EMULATION)
/* $(NETWORKS) */
net/network.o
/* $(LIBS) */
+ from arch/i386/Makefile
    arch/i386/lib/lib.a
lib/lib.a
--end-group
-o vmlinux
$(NM) vmlinux | grep ... | sort > System.map
tags: vim için linux/tags üretir
modules: modülleri üretir
modules_install: modülleri kurar
clean mrproper distclean: derlemenin yapıldığı dizini temizler
psdocs pdfdocs htmdocs mandocs: çekirdek belgelerini üretir

include Rules.make

rpm: bir rpm üretir
```

Burada `--start-group` ve `--end-group`, sembol referans sorununu çözmek için **ld** komut satırı seçenekleridir. Ayrıntılar için [Using LD, the GNU linker: Command Line Options<sup>\(B8\)</sup>](#) belgesine başvurabilirsiniz.

`Rules.make` çoklu Makefile dosyaları için kurallar içerir.

## 2.2. linux/arch/i386/vmlinux.lds

Derlemeden sonra **ld** birkaç nesne ve arşiv dosyasını biraraya getirir, verilerini yeniden yerleştirir ve sembol referansları bağlayıp kapatır. `linux/arch/i386/vmlinux.lds` dosyası `linux/Makefile` tarafından yerleşik çekirdek görüntüsü `linux/vmlinux` 'un ilintilenmesinde kullanılan ilintileyici betik olarak tasarlanmıştır.

```
/* i386 Linux çekirdeği oluşturmak için ld betiği
 * Martin Mares <mj@atrey.karlin.mff.cuni.cz> tarafından yazılmıştır;
 */
OUTPUT_FORMAT("elf32-i386", "elf32-i386", "elf32-i386")
OUTPUT_ARCH(i386)
/* "ENTRY" linux/Makefile içinde "-e stext" komut satırı seçeneği ile değiştirilir */
ENTRY(_start)
/* Çıktı dosyası (linux/vmlinux) yerleşimi.
 * Bakınız Using LD, the GNU linker: Specifying Output Sections (B9) */
SECTIONS
{
/* Çıktı bölümü .text 3G+1M adresinden başlar.
 * Bakınız Using LD, the GNU linker: The Location Counter (B10) */
. = 0xC0000000 + 0x100000;
_text = .; /* Metin ve salt okunur veri */
.text: {
    *(.text)
    *(.fixup)
    *(.gnu.warning)
} = 0x9090
/* Tahsis edilmemiş oyuklar 0x9090 ile doldurulur, örn. "NOP NOP" için opcode.
 * Bakınız Using LD, the GNU linker: Optional Section Attributes (B11) */

_etext = .; /* text bölümünün sonu */

.rodata: { *(.rodata) *(.rodata.*) }
.kstrtab: { *(.kstrtab) }

/* Sonraki 16 baytlık sınıra hizalandı.
 * Bakınız Using LD, the GNU linker: Arithmetic Functions (B12) */
. = ALIGN(16); /* Olağandışılık tablosu */
__start__ex_table = .;
__ex_table: { *(__ex_table) }
__stop__ex_table = .;

__start__ksymtab = .; /* Çekirdek sembol tablosu */
__ksymtab: { *(__ksymtab) }
__stop__ksymtab = .;

.data: { /* Veri */
    *(.data)
    CONSTRUCTORS
}
/* "CONSTRUCTORS" için bakınız:
 * Using LD, the GNU linker: Option Commands (B13) */
_edata = .; /* data bölümünün sonu */
```

```
. = ALIGN(8192); /* init_task */
.data.init_task: { *(.data.init_task) }

. = ALIGN(4096); /* Init kodu ve veri */
__init_begin = .;
.text.init: { *(.text.init) }
.data.init: { *(.data.init) }
. = ALIGN(16);
__setup_start = .;
.setup.init: { *(.setup.init) }
__setup_end = .;
__initcall_start = .;
.initcall.init: { *(.initcall.init) }
__initcall_end = .;
. = ALIGN(4096);
__init_end = .;

. = ALIGN(4096);
.data.page_aligned: { *(.data.idt) }

. = ALIGN(32);
.data.cacheline_aligned: { *(.data.cacheline_aligned) }

__bss_start = .; /* BSS */
.bss: {
    *(.bss)
}
_end = . ;

/* Çıktı bölümü /DISCARD/ son ilintileme çıktısına dahil edilmeyecektir.
 * Bakınız Using LD, the GNU linker: Section Definitions (B14) */
/* Çıkarılacak bölümler */
/DISCARD/: {
    *(.text.exit)
    *(.data.exit)
    *(.exitcall.exit)
}

/* Aşağıdaki çıktı bölümleri 0. bellek konumuna adreslenmiştir.
 * Bakınız Using LD, the GNU linker: Optional Section Attributes (B15) */
/* Stab hata ayıklama bölümleri. */
.stab 0: { *(.stab) }
.stabstr 0: { *(.stabstr) }
.stab.excl 0: { *(.stab.excl) }
.stab.exclstr 0: { *(.stab.exclstr) }
.stab.index 0: { *(.stab.index) }
.stab.indexstr 0: { *(.stab.indexstr) }
.comment 0: { *(.comment) }
}
```

## 2.3. linux/arch/i386/Makefile

linux/arch/i386/Makefile, linux/Makefile tarafından i386'ya özel öğelerin ve kuralların sağlanması için dahil edilmiştir.

Aşağıdaki bütün hedefler linux/Makefile'ının `vmlinux` hedefine bağımlıdır. Bunlar, `linux/arch/i386/boot/Makefile` içinde bazı seçenekler ile ilgili hedefler oluşturarak yapılır.

#### linux/arch/i386/Makefile içindeki hedefler

Hedef	Komut
zImage <sup>(1)</sup>	@\$(MAKE) -C arch/i386/boot zImage <sup>(2)</sup>
bzImage	@\$(MAKE) -C arch/i386/boot bzImage
zlilo	@\$(MAKE) -C arch/i386/boot BOOTIMAGE=zImage zlilo
bzlilo	@\$(MAKE) -C arch/i386/boot BOOTIMAGE=bzImage zlilo
zdisk	@\$(MAKE) -C arch/i386/boot BOOTIMAGE=zImage zdisk
bzdisk	@\$(MAKE) -C arch/i386/boot BOOTIMAGE=bzImage zdisk
install	@\$(MAKE) -C arch/i386/boot BOOTIMAGE=bzImage install

Bu makefile'ın `linux/Makefile` tarafından ihraç edilen bazı çevre değişkenlerini yeniden tanımlaması önemlidir. Özellikle:

```
OBJCOPY=$(CROSS_COMPILE)objcopy -O binary -R .note -R .comment -S
```

Etki, alt dizindeki makefile'lara geçecek ve aracın davranışı değişecektir. `objcopy` komut satırı seçeneklerinin ayrıntıları için [GNU Binary Utilities: objcopy<sup>\(B18\)</sup>](#)'ye bakınız.

`$(LIBS)` 'in niçin "`$(TOPDIR)/arch/i386/lib/lib.a`"yı iki defa içerdiği belli değildir.

```
LIBS := $(TOPDIR)/arch/i386/lib/lib.a $(LIBS) $(TOPDIR)/arch/i386/lib/lib.a
```

Bazı araç dizileri ile ortaya çıkan ilintileme sorunları üzerinde çalıştırmak için olabilir.

## 2.4. linux/arch/i386/boot/Makefile

`linux/arch/i386/boot/Makefile` ne `linux/arch/i386/Makefile` içerisine ne de `linux/Makefile` içerisine dahil edilmediği için biraz daha bağımsızdır.

Bununla birlikte biraz ilişki vardır:

- `linux/Makefile`: yerleşik çekirdek görüntüsü `linux/vmlinux`'u üretir;
- `linux/arch/i386/boot/Makefile`: önyükeme kodunu üretir;
- `linux/arch/i386/Makefile`: `linux/vmlinux`'un önyükeme kodu oluşturulmadan önce hazır olduğunu kontrol eder ve hedefleri `linux/Makefile`'a ihraç eder (`bzImage` gibi).

`zdisk`, `zlilo` veya `zdisk` hedefleri için gereken `$(BOOTIMAGE)` değeri `linux/arch/i386/Makefile`'dan gelir.

#### linux/arch/i386/boot/Makefile içindeki hedefler

Hedef	Komut
zImage	\$(OBJCOPY) compressed/vmlinux compressed/vmlinux.out tools/build bootsect setup compressed/vmlinux.out \$(ROOT_DEV) > zImage
bzImage	\$(OBJCOPY) compressed/bvmlinux compressed/bvmlinux.out tools/build -b bbootsect bsetup compressed/bvmlinux.out \$(ROOT_DEV) \ > bzImage
zdisk	dd bs=8192 if=\$(BOOTIMAGE) of=/dev/fd0



zilo	<pre> if [ -f \$(INSTALL_PATH)/vmlinuz ]; then mv \$(INSTALL_PATH)/vmlinuz     \$(INSTALL_PATH)/vmlinuz.old; fi if [ -f \$(INSTALL_PATH)/System.map ]; then mv \$(INSTALL_PATH)/System.map     \$(INSTALL_PATH)/System.old; fi cat \$(BOOTIMAGE) &gt; \$(INSTALL_PATH)/vmlinuz cp \$(TOPDIR)/System.map \$(INSTALL_PATH)/ if [ -x /sbin/lilo ]; then /sbin/lilo; else /etc/lilo/install; fi </pre>
install	<pre> sh -x ./install.sh \$(KERNELRELEASE) \$(BOOTIMAGE) \$(TOPDIR)/System.map     "\$(INSTALL_PATH)" </pre>

**tools/build** {bootsect, setup, compressed/vmlinux.out}'dan zImage'ı veya {bbootsect, bsetup, compressed/bvmlinux.out}'dan bzImage'ı derler. linux/Makefile "export ROOT\_DEV = CURRENT". Dikkat ederseniz \$(OBJCOPY) [linux/arch/i386/Makefile](#) (sayfa: 7) içerisindeki linux/arch/i386/Makefile tarafından yeniden tanımlanmıştır.

#### linux/arch/i386/boot/Makefile içindeki hedeflerin desteklenmesi

Hedef: Öngereksinimler	Komut
compressed/vmlinux: linux/vmlinux	@\$(MAKE) -C compressed vmlinux
compressed/bvmlinux: linux/vmlinux	@\$(MAKE) -C compressed bvmlinux
tools/build: tools/build.c	\$(HOSTCC) \$(HOSTCFLAGS) -o \$@ \$< -I\$(TOPDIR)/include <sup>(3)</sup>
bootsect: bootsect.o	\$(LD) -Ttext 0x0 -s --oformat binary bootsect.o <sup>(4)</sup>
bootsect.o: bootsect.s	\$(AS) -o \$@ \$<
bootsect.s: bootsect.S ...	\$(CPP) \$(CPPFLAGS) -traditional \$(SVGA_MODE) \$(RAMDISK) \$< -o \$@
bbootsect: bbootsect.o	\$(LD) -Ttext 0x0 -s --oformat binary \$< -o \$@
bbootsect.o: bbootsect.s	\$(AS) -o \$@ \$<
bbootsect.s: bootsect.S ...	\$(CPP) \$(CPPFLAGS) -D__BIG_KERNEL_ -traditional \$(SVGA_MODE) \$(RAMDISK) \$< -o \$@
setup: setup.o	\$(LD) -Ttext 0x0 -s --oformat binary -e begtext -o \$@ \$<
setup.o: setup.s	\$(AS) -o \$@ \$<
setup.s: setup.S video.S ...	\$(CPP) \$(CPPFLAGS) -D__ASSEMBLY_ -traditional \$(SVGA_MODE) \$(RAMDISK) \$< -o \$@
bsetup: bsetup.o	\$(LD) -Ttext 0x0 -s --oformat binary -e begtext -o \$@ \$<
bsetup.o: bsetup.s	\$(AS) -o \$@ \$<
bsetup.s: setup.S video.S ...	\$(CPP) \$(CPPFLAGS) -D__BIG_KERNEL_ -D__ASSEMBLY_-traditional \$(SVGA_MODE) \$(RAMDISK) \$< -o \$@

Dikkat ederseniz bootsect.S bbootsect.s'ye ve setup.S bsetup.s'ye derlendiğinde -D\_\_BIG\_KERNEL\_\_'e sahipti. Bunların kodları Yerden Bağımsız Kod (Place Independent Code – PIC) olmalı, bu durumda -Ttext seçeneğinin ne yaptığının önemi yoktur.

## 2.5. linux/arch/i386/boot/compressed/Makefile

Bu dosya görüntüsünün sıkıştırılıp çözülmesi mekanizmasını yerine getirir.

Sıkıştırma/çözmeyi önyükeme kodundan ayırmak iyidir. Bu böl-ve-yönet çözümü bizim sıkıştırma/çözme mekanizmasını kolayca geliştirmemize veya yeni bir önyükeme yöntemi uyarlamamıza imkan sağlar.

linux/arch/i386/boot/compressed/ dizini head.S ve misc.c dosyalarını içerir.

### linux/arch/i386/boot/compressed/Makefile içindeki hedefler

Hedef	Komut
vmlinux <sup>(5)</sup>	<code>\$(LD) -Ttext 0x1000 -e startup_32 -o vmlinux head.o misc.o piggy.o</code>
bvmlinux	<code>\$(LD) -Ttext 0x100000 -e startup_32 -o bvmlinux head.o misc.o piggy.o</code>
head.o	<code>\$(CC) \$(AFLAGS) -traditional -c head.S</code>
misc.o	<code>\$(CC) \$(CFLAGS) -DKBUILD_BASENAME=\$(subst \$(comma),_,\$(subst -,_,\$(*F))) -c misc.c<sup>(6)</sup></code>
piggy.o	<pre>tmpiggy=_tmp_\$\$\$piggy; \ rm -f \$\$tmpiggy \$\$tmpiggy.gz \$\$tmpiggy.lnk; \ \$(OBJCOPY) \$(SYSTEM) \$\$tmpiggy; \ gzip -f -9 &lt; \$\$tmpiggy &gt; \$\$tmpiggy.gz; \ echo "SECTIONS { .data: { input_len = .; \       LONG(input_data_end - input_data) input_data = .; \       *(.data) input_data_end = .; }}" &gt; \$\$tmpiggy.lnk; \ \$(LD) -r -o piggy.o -b binary \$\$tmpiggy.gz -b elf32-i386 \       -T \$\$tmpiggy.lnk; \ rm -f \$\$tmpiggy \$\$tmpiggy.gz \$\$tmpiggy.lnk</pre>

piggy.o input\_len değişkenini ve gzip ile sıkıştırılmış linux/vmlinux'ı içerir. input\_len piggy.o'nun başında yer alır ve input\_len'in kendisi hariç piggy.o'nun boyutuna eşittir. piggy.o bağlayıcı betiğinde geniş açıklamalar için bakınız [Using LD, the GNU linker: Section Data Expressions<sup>\(B23\)</sup>](#).

Tam olarak söylemek gerekirse **objcopy** tarafından üretilen gzip ile sıkıştırılmış olan linux/vmlinux'un kendisi (ELF biçiminde) değil onun ikili görüntüsüdür. Dikkat ederseniz \$(OBJCOPY), linux/arch/i386/Makefile [linux/arch/i386/Makefile](#) (sayfa: 7) içerisinde -O binary seçeneğini kullanarak ham ikili çıktı almak için yeniden tanımlanmıştır.

{bootsect, setup} veya {bbootsect, bsetup} bağlandığı zaman, \$(LD) onların ikili biçimde çıktı alınması için --oformat binary seçeneğini belirtir. zImage (veya bzImage) yapıldığında, \$(OBJCOPY) compressed/vmlinux (veya compressed/bvmlinux)'dan da bir ara ikili çıktı üretir. zImage veya bzImage'in tüm bileşenlerinin ham ikili biçimde olması görüntünün kendi kendini bir yükleyiciye başvurmadan yükleyebilmesi ve yerleştirebilmesi için elzemdir.

Hem vmlinux hem de bvmlinux head.o ve misc.o'yu piggy.o'dan önce hazırlar, fakat farklı başlangıç adreslerine (0x1000 ve 0x100000) bağlanırlar.

## 2.6. linux/arch/i386/tools/build.c

linux/arch/i386/tools/build.c zImage veya bzImage üretmek için konak aracıdır.

linux/arch/i386/boot/Makefile içinde:

```
tools/build bootsect setup compressed/vmlinux.out $(ROOT_DEV) > zImage
```

```
tools/build -b bbootsect bsetup compressed/bvmlinux.out $(ROOT_DEV) > bzImage
```

`-b` büyük\_çekirdek\_mi (is\_big\_kernel) anlamındadır, sistem görüntüsünün çok büyük olup olmadığını kontrol eder.

**tools/build** aşağıdaki zImage veya bzImage'a yeniden yönlendirilen çıktıları standart çıktı birimine verir:

1. `bootsect` veya `bbootsect`: `linux/arch/i386/boot/bootsect.S`'den, 512 bytes;
2. `setup` veya `bsetup`: `linux/arch/i386/boot/setup.S`'den, 4 sektör veya daha fazla, sektör hizalı;
3. aşağıdakileri de içeren, `compressed/vmlinux.out` veya `compressed/bvmlinux.out`:
  - a. `head.o`: `linux/arch/i386/boot/compressed/head.S` dosyasından;
  - b. `misc.o`: `linux/arch/i386/boot/compressed/misc.c` dosyasından;
  - c. `piggy.o`: `input_len` ve `gzip`'li `linux/vmlinux` dosyasından.

**tools/build** standart çıktıya yazarken `bootsect` veya `bbootsect`'in bazı içeriklerini değiştirecektir:

#### tools/build tarafından yapılan değişiklikler

Offset	Byte	Değişken	Komut
1F1 (497)	1	setup_sectors	kurulum sektörleri sayısı, >=4
1F4 (500)	2	sys_size	system size in 16-bytes cinsinden sistem boyutu, küçük-biten
1FC (508)	1	minor_root	root dev minor
1FD (509)	1	major_root	root dev major

Takip eden bölümde `compressed/vmlinux` `vmlinux` olarak ve `compressed/bvmlinux` `bvmlinux` olarak anılacaktır.

## 2.7. Kaynakça

- Linux Kernel Makefiles: `linux/Documentation/kbuild/makefiles.txt`
- The Linux Çekirdeği NASIL<sup>(B25)</sup>
- GNU make<sup>(B26)</sup>
- Using LD, the GNU linker<sup>(B27)</sup>
- GNU Binary Utilities<sup>(B28)</sup>
- GNU Bash<sup>(B29)</sup>

## 3. linux/arch/i386/boot/bootsect.S

Verildiği gibi `bbootsect`, `bsetup` ve `bvmlinux`'dan oluşan (`head.o`, `misc.o`, `piggy.o`), `bzImage`'i yüklüyoruz. İlk floppy sektörü, `bbootsect` (512 bytes) `linux/arch/i386/boot/bootsect.S`'den derlenmiştir. BIOS tarafından 07C0:0'a yüklenir. `bzImage`'in sıfırlaması (`bsetup` ve `bvmlinux`) henüz yüklenmiş değildir.

### 3.1. Bootsect'i Taşı

```
SETUPSECTS = 4 /*varsayılan kurulum sektörü sayısı*/
BOOTSEG = 0x07C0 /*önyükeme sektörünün orjinal adresi*/
INITSEG = DEF_INITSEG (0x9000) /*önyüklemeyi buraya taşıyoruz;yolun dışına*/
```

```

SETUPSEG  = DEF_SETUPSEG (0x9020) /*kurulum buradan başlar*/
SYSSEG    = DEF_SYSSEG   (0x1000) /*sistem 0x10000 (65536) adresine yüklendi*/
SYSSIZE   = DEF_SYSSIZE  (0x7F00) /*sistem boyutu: 16-bayt sayısı*/
                                     /*yüklenecek*/
ROOT_DEV  = 0              /*ROOT_DEV şimdi "build" tarafından yazıldı*/
SWAP_DEV  = 0              /*SWAP_DEV şimdi "build" tarafından yazıldı*/

.code16
.text

////////////////////////////////////
_start:
{
    // kendimizi 0x7C00'den 0x90000'e taşıdık ve oraya sıçradık.
    move BOOTSEG:0 to INITSEG:0 (512 bytes);
    goto INITSEG:go;
}

```

bbootsect INITSEG:0 (0x9000:0)'a taşınmıştır. Artık BOOTSEG'i unutabiliriz.

### 3.2. Disk parametrelerini Al

```

////////////////////////////////////
// yığıt ve disk parametre tablosunu hazırla
go:
{
    SS:SP = INITSEG:3FF4; // yığıt INITSEG:0x4000-12'ye koy
    /* 0x4000 keyfi bir değerdir >=
       * bootsect boyutu + setup boyutu + yığıt için oda;
       * 12 disk parametre boyutudur. */
    disk parametresini (0:0078'deki gösterici) \
    INITSEG:3FF4 adresine kopyala (12 bytes);
    // int1E: SYSTEM DATA - DISKETTE PARAMETERS(B30)
    yama sektör sayısı 36'ya (parametre tablosunda 4. konum, 1 byte);
    disk parametre tablosu göstericisine (0:0078, int1E) INITSEG:3FF4 ata;
}

```

SS kaydedicisinden hemen sonra SP'nin başlatıldığından emin olun. SS'nin tavsiye edilen değişiklik yöntemi lss komutunu kullanmaktır (IA-32 Intel Architecture Software Developer's Manual<sup>(B31)</sup>, (Vol.3. Ch.5.8.3. Masking Exceptions and Interrupts When Switching Stacks).

İtme ve çekme gibi yığıt işlemleri artık kabul edilebilir. Önce disk parametrelerinin 12 byte'ı INITSEG:3FF4'e kopyalanmıştı.

```

////////////////////////////////////
// disk sürücü parametrelerini al, özellikle sektör/iz sayısı.
char disksizes[] = {36, 18, 15, 9};
int sectors;
{
    SI = disksizes; // i = 0;
    do {
probe_loop:
        sectors = DS:[SI++]; // sectors = disksizes[i++];
        if (SI>=disksizes+4) break; // if (i>=4) break;
        int13/AH=02h(AL=1, ES:BX=INITSEG:0200, CX=sectors, DX=0);
        // int13/AH=02h: DISK - READ SECTOR(S) INTO MEMORY(B32)
    } while (sektör okuma hatası);
}

```

```
}

```

"lods" DS:[SI]'dan AL'ye baytları yükler ve SI'yi otomatik olarak artırır.

İz başına düşen sektör sayısı `sectors` değişkenine kaydedilir.

### 3.3. Kurulum Kodunu Yükle

`bsetup` (`setup_sects` sektörü) `bbootsect`'den hemen sonra yüklenecektir, örn. `SETUPSEG:0`. Dikkat ederseniz `INITSEG:0200==SETUPSEG:0` ve `setup_sects` **tools/build** tarafından `bsetup` boyutunun [linux/arch/i386/tools/build.c](#) (sayfa: 10)'dekiyle eşleşmesi için değiştirilmiştir.

```
////////////////////////////////////
got_sectors:
    word sread;                // geçerli iz için sektörlerin okunması
    char setup_sects;          // tools/build tarafından üzerine yazılmış
{
    print out "Loading";
    /* int10/AH=03h(BH=0): VIDEO - İMLEÇ KONUMUNU VE BOYUTUNU AL (B34)
       * int10/AH=13h(AL=1, BH=0, BL=7, CX=9, DH=DL=0, ES:BP=INITSEG:$msg1): (B35)
       * VIDEO - DİZGEYİ YAZ (B36) */

    // kurulum-sektörlerini taşınan (0x90200 adresine) önyükeme bloğundan
    // (bootblock) sonra doğrudan yükle.
    SI = &sread;                // sread, head ve track indekslemek için SI kullanımı
    sread = 1;                  // önyükeme sektörü okundu

    int13/AH=00h(DL=0);         // reset FDC (B37)

    BX = 0x0200;                // bsetup'ı bbootsect'den (512 bytes) hemen sonra oku
    do {
next_step:
        /* silindir çapraz okumayı (cylinder crossing reading) engellemek için,
           * bu sefer kaç tane sektörün oknacağını hesapla */
        uint16 pushw_ax = AX = MIN(sectors-sread, setup_sects);
no_cyl_crossing:
        read_track(AL, ES:BX);    // AX değiştirilmez
        // ES:BX, sread, head and track'e read_track() için değer ata
        set_next(AX);
        setup_sects -= pushw_ax;    // kalanlar - sonraki adım
    } while (setup_sects);
}

```

SI `sread`, `head` ve `track` değişkenlerini indekslemek için `sread` adresine atanmıştır. Onlar bellekte art arda bulunur. `read_track()` ve `set_next()` yöntemlerinin ayrıntıları için bakınız [Diski Oku](#) (sayfa: 14)

### 3.4. Sıkıştırılmış İmgeyi Yükle

`bvmlinux` (`head.o`, `misc.o`, `piggy.o`) `0x100000` adresine yüklenecektir, `syssize*16` bayt.

```
////////////////////////////////////
// vmlinux/bvmlinux'u yükle (head.o, misc.o, piggy.o)
{
    read_it(ES=SYSSEG);
    kill_motor();                // disket sürücü motorunu kapat
    print_nl();                  // CR LF yazar
}

```

`read_it()` ayrıntıları için bakınız [Diski Oku](#) (sayfa: 14). Eğer `zImage`'ı yüklüyorsak, `vmlinux` `0x10000` (`SYSSEG:0`) adresine yüklenir.

`bzImage` (`bbootsect`, `bsetup`, `bvmlinux`) artık tümüyle bellekte bulunuyor.

### 3.5. Ayarlara Git

```

////////////////////////////////////
// hangi kök aygıtın kullanılacağına bak ve setup.S'e sıçra
int root_dev;                // tools/build tarafından üzerine yazılır
{
    if (!root_dev) {
        switch (sectors) {
            case 15: root_dev = 0x0208;    // /dev/ps0 - 1.2Mb
                break;
            case 18: root_dev = 0x021C;    // /dev/PS0 - 1.44Mb
                break;
            case 36: root_dev = 0x0220;    // /dev/fd0H2880 - 2.88Mb
                break;
            default: root_dev = 0x0200;    // /dev/fd0 - auto detect
                break;
        }
    }
}

// önyükeme bloğundan sonra doğrudan yüklenen ayar yordamına sıçra
goto SETUPSEG:0;
}

```

Denetimi `bsetup`'a geçirir. [linux/arch/i386/boot/setup.S](#) (sayfa: 18) bölümündeki `linux/arch/i386/boot/setup.S:start`'a bakınız.

### 3.6. Diski Oku

Aşağıdaki işlev `bsetup` ve `bvmlinux`'u diskten yüklemek için kullanılır. Dikkat ederseniz `syssize` [linux/arch/i386/tools/build.c](#) (sayfa: 10) içindeki `tools/build` tarafından değiştirilmiştir.

```

sread: .word 0                # geçerli iz'in (track) sektör okuması
head:  .word 0                # geçerli kafa
track: .word 0                # geçerli iz
////////////////////////////////////
// load the system image at address SYSSEG:0
read_it(ES=SYSSEG)
    int syssize;                /* 16-bayt türünden sistem boyutu
                                * tools/build tarafından üzerine yazıldı */
{
    if (ES & 0x0fff) die;        // hizalama 64KB değil

    BX = 0;
    for (;;) {
rp_read:
#ifdef __BIG_KERNEL__
        bootsect_helper(ES:BX);
        /* INITSEG:0220==SETUPSEG:0020 - bootsect_kludge,
         * SETUPSEG:bootsect_helper() gösterici içerir.
         * Bu işlev bazı veriyapılarını başlangıç durumuna getirir

```

```

    *   ilk sefer çağırırmda,
    *   ve SYSSEG:0'dan 0x100000'a taşır, her seferinde 64KB,
    *   aşağıdaki çağırırmda.
    *   Bakınız Bootsect Yardımcısı (sayfa: 16). */
#else
    AX = ES - SYSSEG + ( BX >> 4); // kaç tane 16-bayt okuma
#endif
    if (AX > syssize) return;      // herşey yüklendi
ok1_read:
    /* bu sefer uygun AL (okunacak sektörler) al
    *   çapraz silindir okumasını ve BX taşmasını önlemek için. */
    AX = sectors - sread;
    CX = BX + (AX << 9);           // 1 sector = 2^9 bytes
    if (CX overflow && CX!=0) {     // > 64KB
        AX = (-BX) >> 9;
    }
ok2_read:
    read_track(AL, ES:BX);
    set_next(AX);
}

////////////////////////////////////
// diski parametrelerle oku (sread, track, head)
read_track(AL sektörler, ES:BX hedef)
{
    for (;;) {
        printf(".");
        // int10/AH=0Eh: VIDEO - TELETYPE ÇIKTI (B43)

        // sread, track, head) değerlerine göre CX, DX değerlerini ata
        DX = track;
        CX = sread + 1;
        CH = DL;

        DX = head;
        DH = DL;
        DX &= 0x0100;

        int13/AH=02h(AL, ES:BX, CX, DX);
        // int13/AH=02h: DISK - SEKTÖRLERİ BELLEĞE OKU (B44)
        if (disk okuma başarılı) return;
        // "addw $8, %sp" önceki 4 "pushw" işlemini iptal etmek için.
bad_rt:
        print_all();                // yazma hata kodu, AX, BX, CX ve DX
        int13/AH=00h(DL=0);         // reset FDC (B45)
    }
}

////////////////////////////////////
// set ES:BX, sread, head and track for next read_track()
set_next(AX sektörler)
{
    CX = AX;                        // sektörleri oku
    AX += sread;
    if (AX==sectors) {
        head = 1 ^ head;           // head'i 0 ve 1 arasında değiştir
    }
}

```

```

        if (head==0) track++;
ok4_set:
    AX = 0;
}
ok3_set:
    sread = AX;
    BX += CX && 9;
    if (BX overflow) {                // > 64KB
        ES += 0x1000;
        BX = 0;
    }
set_next_fn:
}

```

### 3.7. Bootsect Yardımcısı

`setup.S:bootsect_helper()` sadece `bootsect.S:read_it()` tarafından kullanılır.

`bbootsect` ve `bsetup` yarı ayrı bağlandıkları için kendi kod/veri bölütlerine bağlı konumlar kullanırlar. Değişik bölütlerde `bootsect_helper()` için "call far" (lcall) çağırmak zorundayız ve "return far" dönmeli. Bu çağırma, CS!=DS durumunu yaratan CS değişikliği ile sonuçlanır ve `setup.S` içindeki değişkenleri belirtmek için bölütü niteleyici kullanmak zorundayız.

```

////////////////////////////////////
// bzImage yüklendiğinde bootsect yükleyici tarafından çağırılır
bootsect_helper(ES:BX)
    bootsect_es = 0;                // setup.S içinde tanımlı
    type_of_loader = 0;            // setup.S içinde tanımlı
{
    if (!bootsect_es) {            // ilk sefer için çağırılır
        type_of_loader = 0x20;    // bootsect-yükleyici, version 0
        AX = ES >> 4;
        *(byte*)&bootsect_src_base+2 = AH;
        bootsect_es = ES;
        AX = ES - SYSSEG;
        return;
    }
bootsect_second:
    if (!BX) {                    // 64KB full
        // SYSSEG:0'dan hedefe taşı, her seferinde 64KB
        int15/AH=87h(CX=0x8000, ES:SI=CS:bootsect_gdt);
        // int15/AH=87h: SİSTEM - GENİŞLETİLMİŞ BELLEĞE KOPYALA (B46)
        if (kopyalama hatası) {
            bootsect_panic() {
                prtstr("INT15 refuses to access high mem, giving up.");
bootsect_panic_loop:            goto bootsect_panic_loop;    // never return
            }
        }
        ES = bootsect_es;        // ES'i daima 0x10000 noktasına ata
        *(byte*)&bootsect_dst_base+2)++;
    }
bootsect_ex:
    // AX içindeki taşınmış çerçeveler (16-bayt)
    AH = *(byte*)&bootsect_dst_base+2) << 4;
    AL = 0;
}

```



```

////////////////////////////////////
// data used by bootsect_helper()
bootsect_gdt:
    .word    0, 0, 0, 0
    .word    0, 0, 0, 0

bootsect_src:
    .word    0xffff

bootsect_src_base:
    .byte    0x00, 0x00, 0x01          # base = 0x010000
    .byte    0x93                      # typbyte
    .word    0                        # limit16,base24 =0

bootsect_dst:
    .word    0xffff

bootsect_dst_base:
    .byte    0x00, 0x00, 0x10          # base = 0x100000
    .byte    0x93                      # typbyte
    .word    0                        # limit16,base24 =0
    .word    0, 0, 0, 0                # BIOS CS
    .word    0, 0, 0, 0                # BIOS DS

bootsect_es:
    .word    0

bootsect_panic_mess:
    .string  "INT15 refuses to access high mem, giving up."

```

`type_of_loader` değerinin değiştiğine dikkat edin. Bu konu [Yükleyici Türünü Kontrol Et](#) (sayfa: 22) bölümünde açıklanmıştır.

### 3.8. Muhtelif

Kalanlar destek işlevleridir; değişkenler ve "gerçek kip çekirdek başlığı" parçaları. Dikkat ederseniz `.text` bölümündeki veri kod olarak bulunur, böylece yüklendiğinde uygun bir şekilde başlangıç durumuna getirilebilir.

```

////////////////////////////////////
// bazı küçük işlevler
print_all(); /* hata kodu yaz, AX, BX, CX and DX */
print_nl();  /* CR LF yaz*/
print_hex(); /* SS:BP tarafından gösterilen kelimeyi onaltılık olarak yaz*/
kill_motor() /* disket sürücü motorunu kapat */
{
    #if 1
        int13/AH=00h(DL=0);      // reset FDC(B48)
    #else
        outb(0, 0x3F2);          // outb(val, port)
    #endif
}

////////////////////////////////////
sectors:      .word 0
disksizes:    .byte 36, 18, 15, 9
msg1:         .byte 13, 10
              .ascii "Loading"

```

"gerçek kip çekirdek başlığı"nın bir parçası olan bootsect taşıyıcı 497. konumdan başlar.

```
.org 497
setup_sects:    .byte SETUPSECS          // tools/build tarafından üzerine yazılır
root_flags:    .word ROOT_RDONLY
syssize:       .word SYSSIZE             // tools/build tarafından üzerine yazılır
swap_dev:      .word SWAP_DEV
ram_size:      .word RAMDISK
vid_mode:      .word SVGA_MODE
root_dev:      .word ROOT_DEV           // tools/build tarafından üzerine yazılır
boot_flag:     .word 0xAA55
```

Bu "başlık" `linux/Documentation/i386/boot.txt` içindeki yerleşim örneğiyle uyuşmalıdır.

Konum	Proto	İsim	Anlam
/Boyut			
01F1/1	ALL	setup_sects	setup'ın sektör cinsinden boyutu
01F2/2	ALL	root_flags	Sıfırdan farklı ise, kök dizin salt okunur olarak bağlanır
01F4/2	ALL	syssize	KULLANMAYIN - sadece bootsect.S kullanımı için
01F6/2	ALL	swap_dev	KULLANMAYIN - atıl oldu
01F8/2	ALL	ram_size	KULLANMAYIN - sadece bootsect.S kullanımı için
01FA/2	ALL	vid_mode	Video kip kontrolü
01FC/2	ALL	root_dev	Varsayılan kök aygıt sayısı
01FE/2	ALL	boot_flag	0xAA55 sihirli numara

### 3.9. Kaynakça

- THE LINUX/I386 BOOT PROTOCOL: `linux/Documentation/i386/boot.txt`
- IA-32 Intel Architecture Software Developer's Manual<sup>(B49)</sup>
- Ralf Brown's Interrupt List<sup>(B50)</sup>

<IA-32 Intel Mimarisi Yazılım Geliştiricisinin Kılavuzu> belgesine bu belgede bir çok kez başvurulduğundan kısaca "IA-32 Kılavuzu" diyeceğim.

## 4. linux/arch/i386/boot/setup.S

`setup.S` sistem verilerinin BIOS'dan alınmasından ve onların uygun bir bellek bölgesine yerleştirilmesinden sorumludur.

GNU GRUB<sup>(B51)</sup> ve LILO<sup>(B52)</sup> gibi diğer önyükleyiciler de `bzImage`'i yükleyebilirler. Bu tür önyükleyiciler `bzImage`'i belleğe yüklemeli ve "gerçek-biçim çekirdek başlıkları"nı kurmalıdırlar, özellikle `type_of_loader`; sonra denetimi `bsetup`'a devreder. `setup.S` şunları varsayar:

- `bsetup` veya `setup` `SETUPSEG:0`'a yüklenmemiş olabilir, örn. denetim `setup.S`'ye geçtiğinde `CS` `SETUPSEG`'e eşit olmayabilir;
- `setup`'ın ilk 4 sektörü `bootsect`'den hemen sonra yüklenir. Sıfırlama `vmlinux`'den önce gelecek şekilde `SYSSEG:0`'da yüklü olabilir, bu varsayım `bsetup`'ta etkili olmaz.

### 4.1. Başlık

```

/* LILO'nun doğru yüklendiğinden emin olmak için imza kelimeleri */
#define SIG1      0xAA55
#define SIG2      0x5A5A

INITSEG  = DEF_INITSEG      # 0x9000, önyüklemeyi buraya taşır, yolun dışına
SYSSEG   = DEF_SYSSEG      # 0x1000, sistem 0x10000'a yüklenir (65536).
SETUPSEG = DEF_SETUPSEG    # 0x9020, bu geçerli bölüttür
                                # ... ve CS'nin önceki içeriği

DELTA_INITSEG = SETUPSEG - INITSEG      # 0x0020

.code16
.text

////////////////////////////////////
start:
{
    goto trampoline();           // sıradaki başlığı atla
}

# Bu setup başlığıdır ve %cs:2 (old 0x9020:2) adresinden başlamalıdır
        .ascii  "HdrS"          # başlık imzası
        .word   0x0203          # başlık sürüm numarası (>= 0x0105)
                                # yoksa eski loadlin-1.5 başarısız olur
realmode_swch: .word   0, 0      # default_switch, SETUPSEG
start_sys_seg: .word   SYSSEG
                .word   kernel_version # çekirdek sürüm dizgesini göstererek
                                # yukarıdaki başlık bölümü
                                # loadlin-1.5 (başlık v1.5) ile uyumlu.
                                # Değiştirmeyin

// kernel_version defined below
type_of_loader: .byte   0        # = 0, eski (LILO, Loadlin,
                                #      Bootlin, SYSLX, bootsect...)
                                # atanmış id değerleri için
                                # bakınız Documentation/i386/boot.txt

# bayraklar, kullanılmayan bitler 0 olmalı, (RFU) loadflags arasındaki bitler
loadflags:
LOADED_HIGH      = 1            # Sıfır değilse, çekirdek yüksek yüklenmiştir
CAN_USE_HEAP     = 0x80        # Sıfır değilse, yükleyici ayrıca setup.S'nin
                                # gerisinde ne kadar alanın yığın (heap)
                                # olarak kullanılacağını tutmak için
                                # heap_end_ptr'ye sahiptir.
                                # Neyin boş olduğunu sadece yükleyici bilir

#ifdef __BIG_KERNEL__
        .byte   0
#else
        .byte   LOADED_HIGH
#endif
setup_move_size: .word   0x8000  # kurulum 0x90000'ye yüklenmediğinde
                                # taşıma boyutu. Çekirdeğe sıçramadan
                                # hemen önce kurulumu 0x90000'a taşıyacağız.

                                # Bununla birlikte geride bıraktığımız
                                # ne kadar yüklenmesi gereken veri
                                # bulunduğunu sadece yükleyici bilir.

```

```
code32_start:                # burada yükleyiciler 32-bit kod
                             # için farklı başlangıç
                             # adresleri koyarlar.

#ifdef __BIG_KERNEL__
    .long    0x1000          # 0x1000 = zImage için varsayılan
#else
    .long    0x100000        # 0x100000 = büyük çekirdek için öntanımlı
#endif
ramdisk_image: .long    0    # yüklü ramdisk görüntüsünün adresi
                             # Burada yükleyici görüntüyü yüklediği
                             # 32-bit adresi koyar. Bu sadece
                             # çekirdek tarafından okunacaktır.

ramdisk_size:  .long    0    # boyutu bayt cinsindendir
bootsect_kludge:
    .word    bootsect_helper, SETUPSEG
heap_end_ptr:  .word    modelist+1024 # (Başlık sürümü 0x0201 veya daha büyük)
                             # buradan sonra kurulum sonuna kadar
                             # boşluklar (özel) kurulum tarafından
                             # yerel yığın amaçları için kullanılabilir.

// modelist .text bölümünün sonundadır
pad1:          .word    0
cmd_line_ptr:  .long    0    # (Başlık sürümü 0x0202 veya daha büyük)
                             # Sıfır değilse, çekirdek komut
                             # satırına 32-bit bir gösterici.
                             # Komut satırı kurulumun başı ile
                             # alçak bellek arasına (0xa0000)
                             # yerleştirilmeli veya okunmadan önce
                             # üzerine yazılmalı. Eğer bu alan
                             # kullanılırsa, 0x90000 bölütüyle ilgili
                             # sihirli birşey kalmaz; kurulum
                             # alçak bellekte 0x10000 veya
                             # daha yüksek herhangi bir
                             # yere yerleştirilebilir.
ramdisk_max:   .long    __MAXMEM-1 # (Başlık sürümü 0x0203 veya daha büyük)
                             # initrd içeriği için en
                             # yüksek güvenli adres
```

**\_\_MAXMEM** tanımlaması `linux/asm-i386/page.h` içindedir:

```
/*
 * 0xC0000000'ın bir __PAGE_OFFSET değeri çekirdeğin bir gigabayt
 * sanal adres boşluğuna sahip olduğu anlamına gelir ki bu da
 * kullanabileceğiniz fiziksel bellek miktarını 950MB'a sınırlar
 */
#define __PAGE_OFFSET          (0xC0000000)

/*
 * Bu kadar adres boşluğu vmalloc() ve iomap() olduğu kadar
 * "fixmap" eşleştirmeleri (mappings) için de tahsis edilir.
 */
#define __VMALLOC_RESERVE      (128 << 20)

#define __MAXMEM                (-__PAGE_OFFSET-__VMALLOC_RESERVE)
```

**\_\_MAXMEM**= 1G – 128M değerini verir.

Bu başlık bazı plan örneklerini takip etmelidir. `linux/Documentation/i386/boot.txt` belgesine başvurun:

Konum	Proto	İsim	Anlamı
/Boyut			
0200/2	2.00+	jump	Sıçrama komutu
0202/4	2.00+	header	Sihirli imza "HdrS"
0206/2	2.00+	version	Desteklene önyükeme protokolü sürümü
0208/4	2.00+	realmode_swth	Önyükeme yükleyici çengeli (hook)
020C/2	2.00+	start_sys	load-low bölütü (0x1000) (modası geçmiş)
020E/2	2.00+	kernel_version	Çekirdek sürüm dizgesi göstericisi
0210/1	2.00+	type_of_loader	Önyükeme yükleyici belirteci
0211/1	2.00+	loadflags	Önyükeme protokolü seçenek bayrakları
0212/2	2.00+	setup_move_size	Yüksek bellek boyutuna taşı (çengellerle kullanılır)
0214/4	2.00+	code32_start	Önyükleyici çengeli
0218/4	2.00+	ramdisk_image	initrd yükleme adresi (önyükleyici tarafından atanır)
021C/4	2.00+	ramdisk_size	initrd boyutu (önyükleyici tarafından atanır)
0220/4	2.00+	bootsect_kludge	KULLANMAYIN - sadece bootsect.S kullanımı için
0224/2	2.01+	heap_end_ptr	kurulum bittikten sonra boş bellek
0226/2	N/A	pad1	Kullanılmaz
0228/4	2.02+	cmd_line_ptr	çekirdek komut satırına 32-bit gösterici
022C/4	2.03+	initrd_addr_max	En yüksek yasal initrd adresi

## 4.2. Kod Bütünlüğünü Kontrol Et

setup kodu sürekli olmayabileceği için önce kod bütünlüğünü kontrol etmeliyiz.

```

////////////////////////////////////
trampoline()
{
    start_of_setup();          // asla dönmez
    .space 1024;
}

////////////////////////////////////
// tüm kodun yüklenip yüklenmediğini görmek için imzayı kontrol et
start_of_setup()
{
    // Bootlin bunun daha önce yapılmasına bağlıdır, bakınız bootlin:technic.doc(B53)
    int13/AH=15h(AL=0, DL=0x81);
    // int13/AH=15h: DİSK - DİSK TÜRÜNÜ AL(B54)

#ifdef SAFE_RESET_DISK_CONTROLLER
    int13/AH=0(AL=0, DL=0x80);
    // int13/AH=00h: DİSK - DİSK SİSTEMİNİ RESETLE(B55)
#endif

    DS = CS;
    // kurulum sonunda imzayı kontrol et
    if (setup_sig1!=SIG1 || setup_sig2!=SIG2) {
        goto bad_sig;
    }
    goto goodsig1;
}

```

```

////////////////////////////////////
// bazı küçük işlevler
prtstr(); /* DS:SI'teki ascii'leri yaz */
prtsp2(); /* çift boşluk yaz */
prtspc(); /* tek boşluk yaz */
prtchr(); /* AL'deki ascii'leri yaz */
beep(); /* CTRL-G yaz, örn. bip */

```

Kod bütünlüğünü doğrulamak için imza kontrol edilir.

İmza bulunmazsa kalan `setup` kodu `SYSSEG:0`'da `vmlinux` öncesinde yer alır.

```

no_sig_mess: .string "No setup signature found ..."

goodsig1:
    goto goodsig;                          // yakın sıçrama yap

////////////////////////////////////
// kalan setup kodunu SYSSEG:0'dan CS:0800'e taşı
bad_sig()
    DELTA_INITSEG = 0x0020 (= SETUPSEG - INITSEG)
    SYSSEG = 0x1000
    word start_sys_seg = SYSSEG;            // kurulum başlığında tanımlı
{
    DS = CS - DELTA_INITSEG;                // INITSEG olarak da bilinir
    BX = (byte) (DS:[497]);                 // örn. setup_sects

    // ilk 4 sektör zaten yüklü
    CX = (BX - 4) << 8;                    // kelime cinsinden kalan kod (2-bayt)
    start_sys_seg = (CX >> 3) + SYSSEG;      // gerçek sistem kodu başlangıcı
    move SYSSEG:0 to CS:0800 (CX*2 bytes);

    if (setup_sig1!=SIG1 || setup_sig2!=SIG2) {
no_sig:
    prtstr("No setup signature found ...");
no_sig_loop:
    hlt;
    goto no_sig_loop;
    }
}

```

`hlt` komutu komut çalıştırmayı durdurur ve işlemciyi halt durumuna getirir. İşlemci halt kipine girildiğini belirtecek şekilde özel bir taşıt dizisi üretir. Etkin bir kesme (NMI dahil) oluştuğunda, işlemci çalıştırmayı `hlt` komutundan sonra devam ettirir ve komut göstericisi, `hlt`'i takip eden komutu göstererek, kesme yöneticisi çağırılmadan önce yığıta kaydedilecektir. Böylece işlemciyi tekrar halt durumuna koymak için `hlt`'tan sonra `jmp` komutuna ihtiyaç duyarız.

`setup` kodu doğru yere taşınmıştır. `start_sys_seg` değişkeni gerçek sistem kodunun başladığı yeri gösterir. Eğer `bad_sig` olmazsa `start_sys_seg`, `SYSSEG` olarak kalır.

### 4.3. Yükleyici Türünü Kontrol Et

Yükleyicinin imgeyle uyumluluğunu olduğunu kontrol et.

```

////////////////////////////////////
good_sig()
    char loadflags;                        // setup başlığı içinde

```

```

char type_of_loader;           // setup başlığı içinde
LOADHIGH = 1
{
    DS = CS - DELTA_INITSEG;    // INITSEG olarak da bilinir
    if ( (loadflags & LOADHIGH) &&!type_of_loader ) {
        // Hata, eski yükleyiciler büyük-çekirdek yüklemeye çalışırlar
        prtstr("Wrong loader, giving up...");
        goto no_sig_loop;      // yukarıda bad_sig()'de tanımlı
    }
}

loader_panic_mess: .string "Wrong loader, giving up..."

```

`bootsect_helper()` tarafından `bvmlinux` yüklendiğinde `type_of_loader`'un 0x20 olarak değiştiğine dikkat edin.

#### 4.4. Bellek Boyutunu

kB cinsinden genişletilmiş bellek boyutunu (1M üzeri) elde etmek için üç değişik bellek saptama şeması dene.

İlk olarak bir bellek haritası oluşturmamızı sağlayan `e820h`'yi dene; sonra 32 bitlik bellek boyutu döndüren `e801h`'yi ve son olarak 0–64M döndüren `88h`'yi dene.

```

////////////////////////////////////
// bellek boyunu al
loader_ok()
    E820NR = 0x1E8
    E820MAP = 0x2D0
{
    // bu işleve girerken, DS = CS-DELTA_INITSEG; INITSEG olarak da bilinir
    (long)DS:[0x1E0] = 0;

#ifdef STANDARD_MEMORY_BIOS_CALL
    (byte)DS:[0x1E8] = 0;           // E820NR

    /* method E820H: bakınız ACPI spec(B56)
     * bellek haritası (from hell). e820h belleği farklı türlerden
     * bir bütün deste olarak sınıflandırılmış şekilde döndürür, ve
     * bellek deliklerine ve herşeye izin verir. Biz bu bellek
     * haritasını tararız ve ilk 32 bellek alanının listesini oluştururuz,
     * [E820MAP]'den döneriz. */
meme820:
    EBX = 0;
    DI = 0x02D0;                  // E820MAP
    do {
jmpe820:
        int15/EAX=E820h(EDX='SMAP', EBX, ECX=20, ES:DI=DS:DI);
        // int15/AX=E820h: GET SYSTEM MEMORY MAP(B57)
        if (failed || 'SMAP'!=EAX) break;
        // if (1!=DS:[DI+16]) continue; // kullanışsız
good820:
        if (DS:[1E8]>=32) break;    // entry# > E820MAX
        DS:[0x1E8]++;             // entry# ++;
        DI += 20;                 // tamponu sonraki için ayarla
again820:
    } while (!EBX)                // bitmedi
bail820:

```

```

/* method E801H:
 * 1k parça boyutuyla bellek boyutu, loadlin karıştırmamak için.
 * 0xe801 bellek boyutunu tamamen farklı bir yerde tutarız
 * çünkü muhtemelen 16 bitten daha uzun olacaktır
 * (1e0 kullanınız çünkü bu Larry Augustine'in alternatif bellek
 * tespit şemasını kullanma yöntemidir ve bu yöntem
 * herşeyi aynı yere yazma konusunda hassastır.) */
meme801:
    stc;                // hatalı BIOSlar için uğraş
    CX = DX = 0;
    int15/AX=E801h;
    /* int15/AX=E801h: >64M YAPILANDIRMALAR İÇİN BELLEK BOYUTUNU AL (B58)
     * AX = K cinsinden 1M ve 16M arasında bellek boyutu (en çok 3C00 = 15MB)
     * BX = genişletilmiş bellek, 16M üzeri, 64K bloklar halinde
     * CX = K cinsinden 1M'dan 16M'a yapılandırılmış bellek
     * DX = 16M üzeri yapılandırılmış bellek, 64K bloklar halinde */
    if (failed) goto mem88;
    if (!CX && !DX) {
        CX = AX;
        DX = BX;
    }
e801usecxdx:
    (long)DS:[0x1E0] = ((EDX & 0xFFFF) << 6) + (ECX & 0xFFFF);    // in K
#endif

mem88: // eski geleneksel yöntem
    int15/AH=88h;
    /* int15/AH=88h: SİSTEM - GENİŞLETİLMİŞ BELLEK BOYUTU (B59)
     * AX = mutlak 100000h adresinden başlayan sürekli kB'ların sayısı */
    DS:[2] = AX;
}

```

## 4.5. Donanım Desteği

Klavye, ekran kartı, harddisk, MCA bus'ı ve işaretleme cihazı gibi donanımların desteğini kontrol et.

```

{
    // klavye tekrarlama oranını en çoğa ayarla
    int16/AX=0305h (BX=0);
    // int16/AH=03h: KEYBOARD - SET TYPEMATIC RATE AND DELAY (B60)

    /* Ekran kiplerini kullanıcıya göstermek için
     * ekran kartını ve parametrelerini kontrol et. */
    video();                // see video.S

    // hd0 ve hd1 verisini al
    hd0 verisini (*int41)'dan CS-DELTA_INITSEG:0080'ya (16 bytes) kopyala;
    // int41: SYSTEM DATA - HARD DISK 0 PARAMETRE TABLO ADRESİ (B61)
    hd1 verisini (*int46)'dan CS-DELTA_INITSEG:0090'ya (16 bytes) kopyala;
    // int46: SYSTEM DATA - HARD DISK 1 PARAMETRE TABLE ADRESİ (B62)
    // hd1 var mı kontrol et
    int13/AH=15h (AL=0, DL=0x81);
    // int13/AH=15h: DISK - DİSK TÜRÜNÜ AL (B63)
    if (failed || AH!=03h) {    // AH==03h eğer harddisk ise
no_disk1:
        temizle CS-DELTA_INITSEG:0090 (16 bytes);
    }
}

```



```

    }
is_disk1:

    // Mikro Kanal veriyolu (Micro Channel-MCA bus) için kontrol et
    CS-DELTA_INITSEG:[0xA0] = 0;    // tablo uzunluğunu 0'a ayarla
    int15/AH=C0h;
    /* int15/AH=C0h: SİSTEM - YAPILANDIRMAYI AL (B64)
       *   ES:BX = ROM yapılandırma tablosu */
    if (failed) goto no_mca;
    ROM yapılandırma tablosunu (ES:BX)'den CS-DELTA_INITSEG:00A0 adresine taşı;
    // CX = (table length<14)? CX:16;    sadece ilk 16 bayt
no_mca:

    // PS/2 noktalama cihazlarını (pointing device) kontrol et
    CS-DELTA_INITSEG:[0x1FF] = 0;    // varsayılan noktalama cihazı 0
    int11h();
    // int11h: BIOS - EKİPMAN LİSTESİNİ AL (B65)
    if (AL & 0x04) {                // fare kuruldu
        DS:[0x1FF] = 0xAA;
    }
}

```

## 4.6. APM Desteği

BIOS APM desteğini kontrol et.

```

#if defined(CONFIG_APM) || defined(CONFIG_APM_MODULE)
{
    DS:[0x40] = 0;                // sürüm = 0 APM BIOS olmadığı anlamına gelir
    int15/AX=5300h(BX=0);
    // int15/AX=5300h: Advanced Power Management v1.0+ - INSTALLATION CHECK (B66)
    if (failed || 'PM'!=BX || !(CX & 0x02)) goto done_apm_bios;
    // (CX & 0x02) 32 bit desteklendiği anlamına gelir
    int15/AX=5304h(BX=0);
    // int15/AX=5304h: Advanced Power Management v1.0+ - DISCONNECT INTERFACE (B67)
    EBX = CX = DX = ESI = DI = 0;
    int15/AX=5303h(BX=0);
    /* int15/AX=5303h: Advanced Power Management v1.0+ (B68)
       *   - CONNECT 32-BIT PROTMODE INTERFACE (B69) */
    if (failed) {
no_32_apm_bios:                // no_32_apm_bios etiketini buraya taşıdım
        DS:[0x4C] &= ~0x0002;    // 32 bit destekleme bitini kaldır
        goto done_apm_bios;
    }
    DS:[0x42] = AX, 32-bit kod bölütü temel adresi;
    DS:[0x44] = EBX, giriş noktası konumu;
    DS:[0x48] = CX, 16-bit kod bölütü temel adresi;
    DS:[0x4A] = DX, 16-bit veri bölütü temel adresi;
    DS:[0x4E] = ESI, APM BIOS kod bölütü uzunluğu;
    DS:[0x52] = DI, APM BIOS veri bölütü uzunluğu;
    int15/AX=5300h(BX=0);        // tekrar kontrol et
    // int15/AX=5300h: Advanced Power Management v1.0+ - INSTALLATION CHECK (B70)
    if (success && 'PM'==BX) {
        DS:[0x40] = AX, APM version;
        DS:[0x4C] = CX, APM flags;
    } else {
apm_disconnect:

```

```

    int15/AX=5304h (BX=0);
    /* int15/AX=5304h: Advanced Power Management v1.0+(B71)
       * - DISCONNECT INTERFACE(B72) */
}
done_apm_bios:
}
#endif

```

## 4.7. Korumalı Kip için Hazırlık

```

// kip seçiciyi çağır
{
    if (realmode_swch) {
        realmode_swch();           // kip seçme çengeli
    } else {
rmodeswch_normal:
        default_switch() {
            cli;                   // hiçbir kesmeye izin yok
            outb(0x80, 0x70);      // NMI etkinleştirmeyi kaldır
        }
    }
rmodeswch_end:
}

// gerekliyse kodu tekrar konumla
{
    (long)code32 = code32_start;
    if (!(loadflags & LOADED_HIGH)) { // düşük yüklü zImage
        // 0x0100 <= start_sys_seg < CS-DELTA_INITSEG
do_move0:
        AX = 0x100;
        BP = CS - DELTA_INITSEG;    // INITSEG olarak da bilinir
        BX = start_sys_seg;
do_move:
        sistem imgesini (start_sys_seg:0 .. CS-DELTA_INITSEG:0)'dan
        0100:0'a taşı;             // her seferinde 0x1000 bayt taşı
    }
end_move:
}

```

code32\_start'ın zImage için 0x1000 adresi ile bzImage için ise 0x100000 adresi ile ilklendirildiğine dikkat edin. code32 değeri denetimin *Korumalı Kipe Geç* (sayfa: 28) içindeki linux/arch/i386/boot/compressed/head.S'e geçirilmesi için kullanılacaktır. zImage önyüklemesi yapılırsa vmlinux'u 0100:0'a taşır; bzImage önyüklemesi yapılırsa bvmlinux start\_sys\_seg:0'da kalır. Taşıma adresi linux/arch/i386/boot/compressed/Makefile içindeki -Ttext seçeneğiyle uyuşmalıdır. Bakınız [linux/arch/i386/boot/compressed/Makefile](#) (sayfa: 9).

Sonra, eğer gerekliyse, kod CS-DELTA\_INITSEG:0'dan (bbootsect ve bsetup) INITSEG:0'a taşınır.

```

DS = CS;           // SETUPSEG olarak da bilinir
// sürüm <=201 ile geriye dönük uyumluluğa ihtiyacımız olup olmadığını kontrol et
if (!cmd_line_ptr && 0x20!=type_of_loader && SETUPSEG!=CS) {
    cli;           // taşınırken kesme olarak yığıtı kullanabilir
    // store new SS in DX
    AX = CS - DELTA_INITSEG;
    DX = SS;
    if (DX>=AX) { // yığıt çerçevesi birlikte taşınacak

```

```

    DX = DX + INITSEG - AX; // i.e. SS-CS+SETUPSEG
}
move_self_1:
/* CS-DELTA_INITSEG:0'dan INITSEG:0'ya taşı (setup_move_size bayt)
 * CS:IP üzerindeki kodun üzerine yazmamak için iki adımda
 * (src < dest) taşı fakat aşağıya doğru ("std") */
move CS-DELTA_INITSEG:move_self_here+0x200
to INITSEG:move_self_here+0x200,
setup_move_size-(move_self_here+0x200) bytes;
// INITSEG:move_self_here+0x200 == SETUPSEG:move_self_here
goto SETUPSEG:move_self_here; // artık CS=SETUPSEG
move_self_here:
move CS-DELTA_INITSEG:0 to INITSEG:0,
move_self_here+0x200 bytes; // goto'dan önce eski CS anlamında
DS = SETUPSEG;
SS = DX;
}
end_move_self:
}

```

Tekrar dikkat edin, `bvmlinux`'u yüklediğinde `bootsect_helper()` tarafından `type_of_loader`'un değeri `0x20` olarak değiştirilmektedir.

## 4.8. A20'yi Etkinleştir

A20 sorunu ve çözümü için [A20 – a pain from the past<sup>\(B75\)</sup>](#)'e başvurun.

```

A20_TEST_LOOPS          = 32    # Bekleme başına adım sayısı
A20_ENABLE_LOOPS        = 255   # deneme için toplam döngü
{
#if defined(CONFIG_MELAN)
// Enable A20. AMD Elan bug fix.
outb(0x02, 0x92);          // outb(val, port)
a20_elan_wait:
while (!a20_test());        // testi geçemedi
goto a20_done;
#endif

a20_try_loop:
// Önce, A20 kapısı olmayan bir sistemde olup olmadığımıza bak.
a20_none:
if (a20_test()) goto a20_done; // testi geçti

// Sonra, BIOS'u (INT 0x15, AX=0x2401) dene
a20_bios:
int15/AX=2401h;
// Int15/AX=2401h: SYSTEM - later PS/2s - ENABLE A20 GATE(B76)
if (a20_test()) goto a20_done; // testi geçti

// Klavye denetleyici üzerinden A20'yi etkinleştirmeye çalış
a20_kbc:
empty_8042();
if (a20_test()) goto a20_done; // BIOS gecikmesi durumunda testi tekrarla
outb(0xD1, 0x64);          // komut yaz
empty_8042();
outb(0xDF, 0x60);          // A20 etkin
empty_8042();

```

```

// a20 gerçekten etkin olana kadar bekle
a20_kbc_wait:
    CX = 0;
a20_kbc_wait_loop:
    do {
        if (a20_test()) goto a20_done; // testi geçti
    } while (--CX)

// Son girişim: "yapılandırma portu A"'yı kullan
outb((inb(0x92) | 0x02) & 0xFE, 0x92);
// yapılandırma portu A etkilenene kadar bekle
a20_fast_wait:
    CX = 0;
a20_fast_wait_loop:
    do {
        if (a20_test()) goto a20_done; // testi geçti
    } while (--CX)

// A20 hala cevap vermiyor. Tekrar ayarlamayı dene.
if (--a20_tries) goto a20_try_loop;
prtstr("linux: fatal error: A20 gate not responding!");
a20_die:
    hlt;
    goto a20_die;
}

a20_tries:
    .byte    A20_ENABLE_LOOPS                // i.e. 255
a20_err_msg:
    .ascii   "linux: fatal error: A20 gate not responding!"
    .byte    13, 10, 0

```

I/O port işlemleri için, [Kaynakça](#) (sayfa: 32) içindeki ilgili başvuru materyallerine bakın.

## 4.9. Korumalı Kipe Geç

Tüm 32 bitlik IA-32 işlemcileri ile kod uyumluluğunu sağlamak için, aşağıdaki korumalı kipe geçmeyi sağlayacak adımları yerine getirin:

1. İlk GDT girdisinde boş tanımlayıcı GDT'yi hazırlayın, bir kod bölütü tanımlayıcı ve bir veri bölütü tanımlayıcı;
2. Maskelenebilir donanım kesmeleri ve NMI dahil kesmeleri etkisiz kılın;
3. `lgdt` asm komutunu kullanarak GDTR yazmacına taban adresini ve GDT sınırını yükleyin;
4. `mov cr0` (Intel 386 ve üstü) veya `lmsw` komutu (Intel 286 ile uyumluluk için) kullanarak CR0 yazmacındaki PE bayrağını etkinleştirin;
5. Ardından bir uzak `jmp` veya bir uzak `call` komutu çalıştırın.

Yığıt normal oku/yaz veri bölütüne yerleştirilebilir, böylece adanmış tanımlayıcıya gerek kalmaz.

```

a20_done:
{
    lidt    idt_48;                // load idt with 0, 0;

    // DS:gdt'yi doğrusal göstericiye dönüştür
    *(long*)(gdt_48+2) = DS << 4 + &gdt;
    lgdt    gdt_48;
}

```

```

// yardımcı işlemciyi sıfırla
outb(0, 0xF0);
delay();
outb(0, 0xF1);
delay();

// kesmeleri yeniden programla
outb(0xFF, 0xA1);      // tüm kesmeleri maskele
delay();
outb(0xFB, 0x21);      // irq2 dışında tüm irq'ları maskele

// korumalı kip!
AX = 1;
lmsw ax;               // makina durumu kelimesi (word), CR0'ın 0'dan 15'e
                       // kadar bitleri sadece PE, MP, EM ve TS bayraklarını
                       // etkiler

goto flush_instr;

flush_instr:
BX = 0;                // bir önyükeme gösteren bayrak
ESI = (CS - DELTA_INITSEG) << 4;    // gerçek kip kod göstericisi
/* NOT: Yüksek yüklenen büyük çekirdekler için
 * jmp'i 0x100000, __KERNEL_CS'ye gereksinim duyarız
 *
 * fakat henüz CS yazmacını yüklemedik,
 * bu yüzden hedef konumun varsayılan boyutu hala 16 bit.
 * Bununla birlikte, bir terim öneki (0x66) kullanarak,
 * CPU bizim 48 bit uzak göstericimizi uygun bir şekilde alır.
 * Bakınız (INTEL 80386 Programmer's Reference Manual,
 * Mixing 16-bit and 32-bit code, page 16-6) */

// __KERNEL_CS: [(uint32*)code32]'e git;
.byte 0x66, 0xea
code32: .long 0x1000      // Korumalı Kip için Hazırlık (sayfa: 26)
                       // içinde üstüne yazılır
.word __KERNEL_CS        // bölüt 0x10
// bakınız linux/arch/i386/boot/compressed/head.S:startup_32
}

```

Uzak `jmp` komutu (0xea) CS kaydedicisini günceller. kalan bölüt kaydedicileri (DS, SS, ES, FS ve GS) içeriği daha sonra yeniden yüklenmelidir. terim boyutu öneki (0x66) `jmp`'i 32 bitlik terim `code32`'ye kadar çalıştırılmasına zorlamak için kullanılır. Terim boyutu öneki ayrıntıları için bakınız: IA-32 Manual (Vol.1. Ch.3.6. Operand-size and Address-size Attributes, and Vol.3. Ch.17. Mixing 16-bit and 32-bit Code).

Denetim `linux/arch/i386/boot/compressed/head.S:startup_32`'ye geçirilir. `zImage` için 0x1000 adresinde; `bzImage` için 0x100000 adresinde. Bakınız: [linux/arch/i386/boot/compressed/head.S](#) (sayfa: 32).

ESI toplanan sistem verilerinin bellek alanını gösterir. 16 bitlik gerçek kip çekirdek kodundan 32 bitlik kısma parametre geçirmek için kullanılır. Ayrıntılar için [linux/Documentation/i386/zero-page.txt](#) dosyasına bakınız.

Daha fazla anahtarlama ayrıntısı için IA-32 Manual Vol.3. (Ch.9.8. Software Initialization for Protected-Mode Operation, Ch.9.9.1. Switching to Protected Mode, and Ch.17.4. Transferring Control Among Mixed-Size Code Segments) belgesine başvurun.

## 4.10. Muhtelif

Kalanlar destek işlevleri ve değişkenleridir.

```
/* linux/Makefile hedefleri tarafından oluşturulan makrolar:
 *   include/linux/compile.h ve include/linux/version.h */
kernel_version: .ascii   UTS_RELEASE
                 .ascii   " ("
                 .ascii   LINUX_COMPILE_BY
                 .ascii   "@"
                 .ascii   LINUX_COMPILE_HOST
                 .ascii   ") "
                 .ascii   UTS_VERSION
                 .byte     0

////////////////////////////////////
default_switch() { cli; outb(0x80, 0x70); } /* Kesmeleri ve NMI'yi iptal et */
bootsect_helper(ES:BX); /* bkz. Bootsect Yardımcısı (sayfa: 16) */

////////////////////////////////////
a20_test()
{
    FS = 0;
    GS = 0xFFFF;
    CX = A20_TEST_LOOPS;           // i.e. 32
    AX = FS:[0x200];
    do {
a20_test_wait:
        FS:[0x200] = ++AX;
        delay();
    } while (AX==GS:[0x210] && --CX);
    return (AX!=GS[0x210]);
    // ZF==0 (i.e. NZ/NE, a20_test!=0) means test passed
}

////////////////////////////////////
// klavye komut kuyruğu boş mu, bak
empty_8042()
{
    int timeout = 100000;

    for (;;) {
empty_8042_loop:
        if (!--timeout) return;
        delay();
        inb(0x64, &AL);           // 8042 durum portu
        if (AL & 1) {               // çıktı
            delay();
            inb(0x60, &AL);         // oku
no_output:} else if (!(AL & 2)) return; // girdi yok
        }
    }

////////////////////////////////////
// CMOS saatini oku, AL'den saniyeyi döndür, video.S'de kullanılır
gettime()
{
    int1A/AH=02h();
}
```

```

/* int1A/AH=02h: SAAT - GERÇEK SAAT ZAMANINI AL (B81)
 * DH = BCD gösterimli saniye*/
AL = DH & 0x0F;
AH = DH >> 4;
aad;
}

//////////////////////////////////////
delay() { outb(AL, 0x80); } // I/O yaptıktan sonra gerekli

// Tanımlayıcı tablo
gdt:
.word 0, 0, 0, 0 # dummy
.word 0, 0, 0, 0 # kullanılmadı
// bölüt 0x10, __KERNEL_CS
.word 0xFFFF # 4Gb - (0x100000*0x1000 = 4Gb)
.word 0 # taban adres = 0
.word 0x9A00 # kodu oku/çalıştır
.word 0x00CF # tanelilik (granularity) = 4096, 386
# (sınırın 5. yarım baytı)

// bölüt 0x18, __KERNEL_DS
.word 0xFFFF # 4Gb - (0x100000*0x1000 = 4Gb)
.word 0 # taban adres = 0
.word 0x9200 # veri oku/yaz
.word 0x00CF # tanelilik = 4096, 386
# (sınırın 5. yarım baytı)

idt_48:
.word 0 # idt sınırı = 0
.word 0, 0 # idt tabanı = 0L
/* [gdt_48] komutla eşleşmesi için 0x0800 (2048) olmalı,
 * Linux 2.2.22'nin yaptığı gibi. */
gdt_48:
.word 0x8000 # gdt sınırı=2048,
# 256 GDT girdisi
.word 0, 0 # gdt tabanı (daha sonra doldurulur)

#include "video.S"

// setup.S'nin sonundaki imza:
{
setup_sig1: .word SIG1 // 0xAA55
setup_sig2: .word SIG2 // 0x5A5A
modelist:
}

```

video.S içindeki video ayar ve algılama kodu:

```

ASK_VGA = 0xFFFFD // defined in linux/include/asm-i386/boot.h
//////////////////////////////////////
video()
{
    pushw DS; // farklı bölütler kullan
    FS = DS;
    DS = ES = CS;
    GS = 0;
    cld;
    basic_detect(); // temel kart türü testi (EGA/VGA/MDA/CGA)
}

```

```

#ifdef CONFIG_VIDEO_SELECT
    if (FS:[0x01FA] != ASK_VGA) {          // kullanıcı seçimli video kipi
        mode_set();
        if (failed) {
            prtstr("You passed an undefined mode number.\n");
            mode_menu();
        }
    } else {
vid2:
        mode_menu();
    }
vid1:
#ifdef CONFIG_VIDEO_RETAIN
    restore_screen();                    // ekran içeriğini geri yükle
#endif /* CONFIG_VIDEO_RETAIN */
#endif /* CONFIG_VIDEO_SELECT */
    mode_params();                      // kip parametrelerini sakla
    popw ds;                            // orjinal DS'yi geri yükle
}

```

/\* YAPILACAKLAR: video() ayrıntıları \*/

#### 4.11. Kaynakça

- [A20 – a pain from the past<sup>\(B82\)</sup>](#)
- [Real-time Programming<sup>\(B83\)</sup>](#) Appendix A: Complete I/O Port List
- [IA-32 Intel Architecture Software Developer's Manual<sup>\(B84\)</sup>](#)
- Summary of empty\_zero\_page layout (kernel point of view):  
[linux/Documentation/i386/zero-page.txt](#)

## 5. linux/arch/i386/boot/compressed/head.S

Artık `bvmlinux` içindeyiz! `misc.c:decompress_kernel()`'in yardımıyla, çekirdek imgemiz `linux/vmlinux`'u elde etmek için `piggy.o`'yu açacağız.

Bu dosya saf 32 bitlik açılış kodudur. Önceki iki dosyadan farklı olarak kaynak kodun içinde hiç `.code16` deyimi yoktur. Ayrıntılar için [Using as: Writing 16-bit Code<sup>\(B85\)</sup>](#) belgesine başvurun.

### 5.1. Sıkıştırılmış Çekirdeğin Açılması

Bölüt tanımlayıcılarındaki (bölüt seçicisi `__KERNEL_CS` ve `__KERNEL_DS`'ye tekabül eden) bölüt taban adresleri 0'a eşittir; bu yüzden, eğer bu iki adresten her ikisi de kullanıldıysa, mantıksal konum adresi (bölüt:konum biçiminde) kendi doğrusal adresine eşit olacaktır. `zImage` için, `CS:EIP` şimdi 10:1000 mantıksal adresinde (doğrusal 0x1000 adresi), `bzImage` ise 10:100000 (doğrusal 0x100000) adresindedir.

Sayfalama etkinleştirilmediği için doğrusal adres fiziksel adres ile özdeştir. Adres konuları için [IA-32 Manual](#) (Vol.1. Ch.3.3. Memory Organization, and Vol.3. Ch.3. Protected-Mode Memory Management) ve [Linux Device Drivers: Memory Management in Linux<sup>\(B86\)</sup>](#) belgelerine bakınız.

`BX=0` ve `ESI=INITSEG<<4` olması `setup.S`'den ileri gelir.

```

.text
////////////////////////////////////
startup_32()

```



```

{
    cld;
    cli;
    DS = ES = FS = GS = __KERNEL_DS;
    SS:ESP = *stack_start; // user_stack[] sonu, misc.c içinde tanımlı
    // korumalı kip etkinleştirildikten sonra
    // tüm bölüt kaydedicileri yeniden yüklenir

    // A20'nin gerçekten etkin olup olmadığını kontrol et
    EAX = 0;
    do {
1:  DS:[0] = ++EAX;
    } while (DS:[0x100000]==EAX);

    EFLAGS = 0;
    clear BSS; // _edata'dan _end'e

    struct moveparams mp; // subl $16,%esp
    if (!decompress_kernel(&mp, ESI)) { // AX'deki değeri döndür
        ESI'yi yığıttan geri yükle;
        EBX = 0;
        goto __KERNEL_CS:100000;
        // bkz. linux/arch/i386/kernel/head.S:startup_32
    }

    /*
    * Yüksek yüklediysek buraya geliriz.
    * move-in-place rutinini aşağı 0x1000'e taşımamız gerekir
    * ve sonra yığıttan aldığımız yazmaçlardaki
    * tampon adresleri ile başlatırız.
    */
3:  move_routine_start..move_routine_end 0x1000'e taşı;
    // move_routine_start & move_routine_end aşağıda tanımlanmıştır

    // move_routine_start() parametrelerini hazırla
    EBX = real mode pointer; // ESI değeri setup.S'den geçer
    ESI = mp.low_buffer_start;
    ECX = mp.lcount;
    EDX = mp.high_buffer_star;
    EAX = mp.hcount;
    EDI = 0x100000;
    cli; // kesme almadığımızdan emin ol.
    goto __KERNEL_CS:1000; // move_routine_start();
}

/* Eğer yüksek yüklediysek, yerinde çözülmüş çekirdeği taşımak için
* yordam (şablon). Bu PIC kodu olmalı! */
////////////////////////////////////
move_routine_start()
{
    mp.low_buffer_start'ı 0x100000'a taşı, mp.lcount bayt,
    iki adımda: (lcount >> 2) kelime + (lcount & 3) bayt;
    move/append mp.high_buffer_start, ((mp.hcount + 3) >> 2) kelime
    // 1 kelime == 4 bayt, 32 bitlik kod/veri anlamında.

    ESI = EBX; // gerçek kip gösterici, setup.S'deki gibi
    EBX = 0;

```

```
goto __KERNEL_CS:100000;
// bkz. linux/arch/i386/kernel/head.S:startup_32()
move_routine_end:
}
```

je 1b ve jnz 3f'nin anlamları için [Using as: Local Symbol Names<sup>\(B87\)</sup>](#) belgesine başvurunuz.

`_edata` ve `_end` tanımlamalarını bulamadınız mı? Sorun değil, onlar "dahili ilintileme betiği" içinde tanımlanmıştır. `-T` (`--script=`) seçeneği belirtilmeksizin kullanılırsa, `ld` bu yerleşik betiği `compressed/bvmlinux`'u ilintilemek için kullanır. Bu betiği görüntülemek için `"ld --verbose"` komutunu kullanınız ya da [Dahili İlintileme Betiği](#) (sayfa: 60) bölümüne bakınız.

`-T` (`--script=`), `-L` (`--library-path=`) ve `--verbose` seçimlerinin tarifi için [Using LD, the GNU linker: Command Line Options<sup>\(B89\)</sup>](#) belgesine başvurunuz. Ayrıca `"man ld"` ve `"info ld"` de yardımcı olabilir.

`piggy.o` çözüldü ve kontrol `__KERNEL_CS:100000`'ye geçirildi, örn.

`linux/arch/i386/kernel/head.S:startup_32()`. Bakınız [linux/arch/i386/kernel/head.S](#) (sayfa: 39).

```
#define LOW_BUFFER_START      0x2000
#define LOW_BUFFER_MAX       0x90000
#define HEAP_SIZE            0x3000
////////////////////////////////////
asmlinkage int decompress_kernel(struct moveparams *mv, void *rmode)
|-- setup_real_mode(=rmode), vidmem, vidport, lines and cols;
|-- if (is_zImage) setup_normal_output_buffer() {
|   output_data      = 0x100000;
|   free_mem_end_ptr = real_mode;
| } else (is_bzImage) setup_output_buffer_if_we_run_high(mv) {
|   output_data      = LOW_BUFFER_START;
|   low_buffer_end   = MIN(real_mode, LOW_BUFFER_MAX) & ~0xffff;
|   low_buffer_size   = low_buffer_end - LOW_BUFFER_START;
|   free_mem_end_ptr = &end + HEAP_SIZE;
|   // get mv->low_buffer_start and mv->high_buffer_start
|   mv->low_buffer_start = LOW_BUFFER_START;
|   /* To make this program work, we must have
|   *   high_buffer_start > &end+HEAP_SIZE;
|   * As we will move low_buffer from LOW_BUFFER_START to 0x100000
|   *   (max low_buffer_size bytes) finally, we should have
|   *   high_buffer_start > 0x100000+low_buffer_size; */
|   mv->high_buffer_start = high_buffer_start
|   = MAX(&end+HEAP_SIZE, 0x100000+low_buffer_size);
|   mv->hcount = 0 if (0x100000+low_buffer_size > &end+HEAP_SIZE);
|   = -1 if (0x100000+low_buffer_size <= &end+HEAP_SIZE);
|   /* mv->hcount==0: we need not move high_buffer later,
|   *   as it is already at 0x100000+low_buffer_size.
|   * Used by close_output_buffer_if_we_run_high() below. */
| }
|-- makecrc();           // create crc_32_tab[]
| puts("Uncompressing Linux... ");
|-- gunzip();
| puts("Ok, booting the kernel.\n");
|-- if (is_bzImage) close_output_buffer_if_we_run_high(mv) {
|   // get mv->lcount and mv->hcount
|   if (bytes_out > low_buffer_size) {
|       mv->lcount = low_buffer_size;
|       if (mv->hcount)
|           mv->hcount = bytes_out - low_buffer_size;
```

```

|         } else {
|             mv->lcount = bytes_out;
|             mv->hcount = 0;
|         }
|     }
| }
|-- return is_bzImage; // return value in AX

```

end "dahili ilintileme betiği" içinde de tanımlanmıştır.

`decompress_kernel()` bir `asmlinkage` değiştiricisine sahiptir. `linux/include/linux/linkage.h` dosyasında:

```

#ifdef __cplusplus
#define CPP_ASMLINKAGE extern "C"
#else
#define CPP_ASMLINKAGE
#endif

#ifdef __i386__
#define asmlinkage CPP_ASMLINKAGE __attribute__((regparm(0)))
#elif defined __ia64__
#define asmlinkage CPP_ASMLINKAGE __attribute__((syscall_linkage))
#else
#define asmlinkage CPP_ASMLINKAGE
#endif

```

`asmlinkage` makrosu derleyiciyi yığıttaki tüm işlev argümanlarını aktarması için (bazı eniyileştirme yöntemleri bunu değiştirmeye çalışsa bile) zorlayacaktır. Ayrıntılar için [Using the GNU Compiler Collection \(GCC\): Declaring Attributes of Functions](#)<sup>(B91)</sup> (`regparm`) ve [Kernelnewbies FAQ: What is asmlinkage](#)<sup>(B92)</sup> belgelerine bakınız.

## 5.2. gunzip()

`decompress_kernel()` sadece `bzImage` için sıkıştırılmış çekirdek imgesini düşük (`output_data` ile gösterilen) ve yüksek (`high_buffer_start` ile gösterilen) tamponlara açmak için `linux/lib/inflate.c` dosyasında tanımlı `gunzip()` -> `inflate()` çağrısını yapar.

gzip dosya biçimi [RFC 1952](#)<sup>(B93)</sup> içinde belirtilmiştir.

### gzip dosya biçimi

Bileşen	Açılımı	Bayt sayısı	Yorumu
ID1	IDentification 1 (1. belirteç)	1	31 (0x1f, \037)
ID2	IDentification 2 (2. belirteç)	1	139 (0x8b, \213) <sup>(7)</sup>
CM	Compression Method (Sıkıştırma Yöntemi)	1	8 – "deflate" sıkıştırma yöntemini gösterir
FLG	FLaGs (Seçenekler)	1	çoğu durumda 0
MTIME	Modification TIME (Değişiklik zamanı)	4	özgün dosyanın değişiklik zamanı
XFL	eXtra FLags (ek seçenekler)	1	2 – sıkıştırıcı en yavaş algoritmayı <sup>(8)</sup> kullanır, azami sıkıştırma yapar
OS	Operating System (İşletim Sistemi)	1	3 – Unix
ek alanlar	–	–	değişken uzunluk, alan FLG ile belirtilir <sup>(9)</sup>
sıkıştırılmış bloklar	–	–	değişken uzunluk
CRC32	–	4	sıkıştırılmamış verinin CRC değeri

ISIZE	Input SIZE (Girdi uzunluğu)	4	sıkıştırılmamış girdi verisi boyunun 2^32 ile bölümünden kalan
-------	-----------------------------	---	----------------------------------------------------------------

Bu dosya biçimi bilgisini gzipli `linux/vmlinuz`'un başlangıcını bulmak için kullanabiliriz.

```
[root@localhost boot]# hexdump -C /boot/vmlinuz-2.4.20-28.9 | grep '1f 8b 08 00'
00004c50  1f 8b 08 00 01 f6 e1 3f 02 03 ec 5d 7d 74 14 55 |.....?...]t.U|
[root@localhost boot]# hexdump -C /boot/vmlinuz-2.4.20-28.9 -s 0x4c40 -n 64
00004c40  00 80 0b 00 00 fc 21 00 68 00 00 00 1e 01 11 00 |.....!.h.....|
00004c50  1f 8b 08 00 01 f6 e1 3f 02 03 ec 5d 7d 74 14 55 |.....?...]t.U|
00004c60  96 7f d5 a9 d0 1d 4d ac 56 93 35 ac 01 3a 9c 6a |.....M.V.5...:j|
00004c70  4d 46 5c d3 7b f8 48 36 c9 6c 84 f0 25 88 20 9f |MF\..{.H6.1..%. .|
00004c80
[root@localhost boot]# hexdump -C /boot/vmlinuz-2.4.20-28.9 | tail -n 4
00114d40  bd 77 66 da ce 6f 3d d6 33 5c 14 a2 9f 7e fa e9 |.wf...o=.3\...~..|
00114d50  a7 9f 7e fa ff 57 3f 00 00 00 00 00 d8 bc ab ea |..~...W?.....|
00114d60  44 5d 76 d1 fd 03 33 58 c2 f0 00 51 27 00      |D]v...3X...Q'.|
00114d6e
```

Yukarıdaki örnekte gzipli dosyanın 0x4c50 adresinde başladığını görebiliriz. "1f 8b 08 00"den önceki dört byte `input_len`'dir (küçük sonlu olarak 0x0011011e) ve 0x4c50+0x0011011e=0x114d6e değeri `bzImage` (/boot/vmlinuz-2.4.20-28.9) dosyasının boyuna eşittir.

```
static uch *inbuf;          /* girdi tamponu */
static unsigned insize = 0; /* inbuf içindeki geçerli baytlar*/
static unsigned inptr = 0;  /* inbuf içinde işlenecek sonraki baytın indisi */
////////////////////////////////////
static int gunzip(void)
{
    Girdi tamponunu {ID1, ID2, CM} için kontrol et, şöyle olmalı:
        {0x1f, 0x8b, 0x08} (normal durum), veya
        {0x1f, 0x9e, 0x08} (gzip 0.5 için);
    FLG'yi (seçenek baytı) kontrol et, 1, 5, 6 ve 7. bitler atanmamalı;
    Ignore {MTIME, XFL, OS};
    FLG biti 2,3 ve 4'e karşılık gelen seçimlik yapıları yönet;
    inflate();                // sıkıştırılmış blokları yönet
    Validate {CRC32, ISIZE};
}
```

`linux/arch/i386/boot/compressed/misc.c` içinde tanımlı `get_byte()` ilk defa çağırıldığında, girdi tamponunu `inbuf=input_data` ve `insize=input_len` olacak şekilde ayarlamak için `fill_inbuf()` işlevini çağırır. `input_data` ve `input_len` sembolleri `piggy.o` ilintileme betiğinde tanımlanmıştır. Bakınız [linux/arch/i386/boot/compressed/Makefile](#) (sayfa: 9).

### 5.3. inflate())

```
// misc.c içindeki bazı önemli tanımlamalar
#define WSIZE 0x8000          /* Pencere boyutu en azından 32k olmalı,
                                * ve ikinin üssü olmalı */
static uch window[WSIZE];    /* Kayan pencere tamponu */
static unsigned outcnt = 0;   /* Çıktı tamponundaki bayt sayısı */

// linux/lib/inflate.c
#define wp outcnt
#define flush_output(w) (wp=(w), flush_window())
STATIC unsigned long bb;     /* bit tamponu */
```

```

STATIC unsigned bk;          /* bit tamponundaki bit sayısı */
STATIC unsigned hufts;       /* belleği kullanımı izlemek*/
static long free_mem_ptr = (long)&end;
////////////////////////////////////
STATIC int inflate()
{
    int e;                   /* son blok seçeneği */
    int r;                   /* sonuç kod */
    unsigned h;              /* struct huft'un azami belleği */
    void *ptr;

    wp = bb = bk = 0;

    // sıkıştırılmış blokları birer birer şişir (inflate)
    do {
        hufts = 0;
        gzip_mark() { ptr = free_mem_ptr; };
        if ((r = inflate_block(&e)) != 0) {
            gzip_release() { free_mem_ptr = ptr; };
            return r;
        }
        gzip_release() { free_mem_ptr = ptr; };
        if (hufts > h)
            h = hufts;
    } while (!e);

    /* Çok fazla ileri bakmayı (lookahead) geri al. Sonraki okuma bayt hizalı
     * olacak böylece son anlamlı bayttaki kullanılmayan bitleri çıkarabileceğiz.
     */
    while (bk >= 8) {
        bk -= 8;
        inptr--;
    }

    /* çıktı penceresini (window[0..outcnt-1]) çıktı verisine (output_data) yaz,
     * output_ptr/output_data, crc ve bytes_out'u da buna bağlı olarak güncelle
     * ve outcnt'yi 0'a ayarla. */
    flush_output(wp);

    /* başarılı olduğunu döndür */
    return 0;
}

```

`free_mem_ptr` dinamik bellek tahsisi için `misc.c:malloc()` içinde kullanılır. Sıkıştırılmış her bir bloğu şişirmeden önce, `gzip_mark()` `free_mem_ptr` değerini saklar. Şişirmeden sonra `gzip_release()` bu değeri geri yükleyecektir. Bu `inflate_block()` içinde ayrılan belleğin serbest bırakılma işlemidir.

**Gzip**<sup>(B95)</sup> dosyaları sıkıştırmak için Lempel-Ziv (LZ77) kodlamasını kullanır. Sıkıştırılmış veri biçimi **RFC 1951**<sup>(B96)</sup> içinde belirtilmiştir. `inflate_block()` bit düzeni olarak ele alınabilen sıkıştırılmış blokları şişirir.

Sıkıştırılmış her bir bloğun veri yapısı anahatlarıyla şöyledir:

```

BFINAL (1 bit)
    0 - son blok değil
    1 - son blok
BTYPE (2 bit)

```

```
00 - sıkıştırma yok
    bayt sınırına kadar kalan bitler;
    LEN      (2 bayt);
    NLEN     (2 bayt, LEN'in tamamlayıcısı);
    data     (LEN bayt);
01 - düzeltilmiş Huffman kodu ile sıkıştırılmış
    {
    literal   (7-9 bitleri, 256 hariç 0..287 kodunu temsil eder);
              // Bakınız RFC 1951, 3.2.6 paragrafındaki tablo.
    length    (0-5 bitleri, literal > 256 ise 3..258 arasında bir uzunluktur);
              // Bkz. RFC 1951, 3.2.5 paragrafındaki 1. alfabe tablosu.
    data      (literal < 256 ise literal baytlarının verileri);
    distance  (literal == 257..285 ise 5 artı 0-13 ek bit,
              1..32768 arasında bir mesafe belirtir;
              /* Bakınız RFC 1951, 3.2.5 paragrafındaki 2. alfabe tablosu,
               * 3.2.6 paragrafındaki deyim değil*/
              /* Çıktı akımında "distance" bayt geri git
               * ve "length" baytı kopyala. */
    }*        // çok sayıda örnek olabilir
    literal   (7 bit, tümü 0, literal == 256, blok sonu belirtir);
10 - Dinamik Huffman koduyla sıkıştırılmış
    HLIT      (5 bit, Literal/Length kodlarının sayısı - 257, 257-286);
    HDIST     (5 bit, Distance kodlarının sayısı - 1, 1-32);
    HCLEN     (4 bit, Code Length kodlarının sayısı - 4, 4 - 19);
    Code Length dizisi ((HCLEN+4)*3 bit)
    /* Aşağıdaki 2 alfabe tablosu, önceki Code Length dizisinden üretilen
     * Huffman kod çözme tablosu kullanılarak çözülecektir. */
    Literal/Length alfabesi (HLIT+257 kod)
    Distance alfabesi       (HDIST+1 kod)
    // Kod çözme tabloları bu alfabe tablolarından oluşturur.
    /* Aşağıdaki, farklı kod çözme tabloları kullanmak dışında düzeltilmiş
     * Huffman kodları kısmı ile benzerlik gösterir. */
    {
    literal/length
        (değişken uzunluk, Literal/Length alfabesine bağımlı);
    data      (literal < 256 ise literal baytlarının verisi);
    distance  (literal == 257..285 ise değişken uzunlukta,
              Distance alfabesine bağımlı);
    }*        // çok sayıda örnek olabilir
    literal   (literal değeri 256, blok sonu anlamında);
11 - reserved (hata)
```

Dikkat ederseniz Huffman kodları MSB'den başlarken, veri elemanları En-Değersiz-Bit'ten (Least-Significant Bit – LSB) başlayıp En-Değerli-Bit'e (Most-Significant Bit – MSB) kadar byte'ları paketler. Ayrıca *literal* değerleri 286-287 ve *distance* kodları 30-31'in asla oluşmayacağına dikkat edin.

RFC 1951 ve yukarıdaki veri yapısı elinizdeyken `inflate_block()` işlevini anlamak çok da zor olmayacaktır. Huffman kodlaması ve alfabe tablosu üretimi için RFC 1951 içindeki ilgili paragraflara başvurunuz.

Daha fazla ayrıntı için `linux/lib/inflate.c` belgesine, gzip kaynak koduna (bir çok yorum satırı var) ve ilgili başvuru materyallerine bakınız.

## 5.4. Kaynakça

- Using `as`<sup>(B97)</sup>
- Using `LD`, the GNU linker<sup>(B98)</sup>
- IA-32 Intel Architecture Software Developer's Manual<sup>(B99)</sup>
- The gzip home page<sup>(B100)</sup>
- gzip (freshmeat.net)<sup>(B101)</sup>
- RFC 1951: DEFLATE Compressed Data Format Specification version 1.3<sup>(B102)</sup>
- RFC 1952: GZIP file format specification version 4.3<sup>(B103)</sup>

## 6. linux/arch/i386/kernel/head.S

Sonunda çekirdek imgesi `linux/vmlinux` yerini aldı. İki girdiye ihtiyaç var:

- 16 bitlik gerçek kip kodunun yerini göstermek için `ESI`; `INITSEG<<4` durumu;
- Hangi işlemcinin çalıştığını belirtmek için `BX`, 0 BSP anlamına gelir, diğer değerler AP için.

`ESI`, daha sonra `empty_zero_page`'e kopyalanacak 16 bitlik gerçek kip kodundaki parametre alanını gösterir. `ESI` sadece BSP için geçerlidir.

BSP (BootStrap Processor) ve APlar (Application Processors) Intel terminolojileridir. Çoklu işlemci (MP) ilk-lendirme işlemleri hakkında bakınız: IA-32 Manual (Vol.3. Ch.7.5. Multiple-Processor (MP) Initialization) ve [MultiProcessor Specification](#)<sup>(B104)</sup>.

Yazılım açısından bakıldığında, çok işlemcili bir sistemde, BSP ve APlar fiziksel belleği paylaşırlar, fakat kendi yazmaç kümelerini kullanırlar. BSP ilk olarak çekirdek kodunu çalıştırır, işletim sistemi çalıştırma ortamını kurar ve APları da onun üzerinde çalışması için tetikler. AP BSP onu uyarıncaya kadar uyku kipinde kalır.

### 6.1. Sayfalamayı Etkinleştir

```
.text
////////////////////////////////////
startup_32()
{
    /* bölütleri bilinen değerlere ata */
    cld;
    DS = ES = FS = GS = __KERNEL_DS;

#ifdef CONFIG_SMP
#define cr4_bits mmu_cr4_features-__PAGE_OFFSET
    /* long mmu_cr4_features linux/arch/i386/kernel/setup.c içinde tanımlı
       * __PAGE_OFFSET = 0xC0000000, örn. 3G */

    // CR4 desteği ile ApP (> Intel 486) CR'ü BSP'den kopyalayacak
    if (BX && cr4_bits) {
        // sayfalamaya seçeneklerini aç (turn on) (PSE, PAE, ...)
        CR4 |= cr4_bits;
    } else
#endif
    {
        /* sayfa tablolarını (pg0..empty_zero_page-1) sadece BSP ilklendirir
           * .org 0x2000'de pg0
           * .org 0x4000'de empty_zero_page
           * toplam (0x4000-0x2000)/4 = 0x0800 girdi */

```

```

pg0 = {
    0x00000007,          // 7 = PRESENT + RW + USER
    0x00001007,          // 0x1000 = 4096 = 4K
    0x00002007,
    ...
pg1:    0x00400007,
    ...
    0x007FF007          // toplam 8M
empty_zero_page:
};
}

```

Bir çekirdek sembolüne başvuracağımız zaman neden `__PAGE_OFFSET` eklemek zorundayız, örneğin `pg0` gibi?

`linux/arch/i386/vmlinux.lds` içinde şunlar bulunur:

```

. = 0xC0000000 + 0x100000;
_text = .;                /* Metin ve salt-okunur veri */
.text : {
    *(.text)
...

```

`linux/vmlinux` için bağlanacak ilk dosya olan `linux/arch/i386/kernel/head.o` içinde, `pg0` `.text` bölümü `0x2000` konumunda olduğu için, çıktı bölümü `.text` içinde `0x2000` konumunda olacaktır. Böylece ilintilendikten sonra `0xC0000000+0x100000+0x2000` adresinde olacaktır.

```

[root@localhost boot]# nm --defined /boot/vmlinux-2.4.20-28.9 | \
grep 'startup_32\|mmu_cr4_features\|pg0\|<empty_zero_page>' | sort
c0100000 t startup_32
c0102000 T pg0
c0104000 T empty_zero_page
c0376404 B mmu_cr4_features

```

Korumalı kipte sayfalama etkinleştirilmeden, doğrusal adres doğrudan fiziksel adrese eşlenecektir. "`movl $pg0-__PAGE_OFFSET,%edi`" `pg0`'ın fiziksel adresine eşit olan `EDI=0x102000` değerini atayacaktır (`linux/vmlinux` `0x100000` adresine yerleştirildiği için). `-PAGE_OFFSET` şeması olmadan, yanlış ve muhtemelen RAM alanının ötesinde olacak `0xC0102000` fiziksel adresine erişecektir.

`mmu_cr4_features` `.bss` bölümü içindedir ve yukarıdaki örnekte `0x376404` fiziksel adresine yerleştirilmiştir.

Sayfa tablosu ilklendirildikten sonra sayfalama etkinleştirilebilir.

```

// sayfa dizini temel göstericisini ata, fiziksel adres
CR3 = swapper_pg_dir - __PAGE_OFFSET;
// sayfalama etkin!
CR0 |= 0x80000000;        // PG bitini ayarla
goto 1f;                  // flush prefetch-queue
1:
EAX = &1f;                // sonraki komutu takip eden adres
goto *(EAX);              // EIP'yi yeniden konumla
1:
SS:ESP = *stack_start;

```

Sayfa dizini `swapper_pg_dir` ([Muhtelif](#) (sayfa: 44) bölümündeki tanımlamalara bakınız), sayfa tabloları `pg0` ve `pg1` ile birlikte, doğrusal `0..8M-1` ve `3G..3G+8M-1` adreslerinin her ikisinin de `0..8M-1` fiziksel adresine eşlendiğini belirtir. Artık çekirdek sembollerine "`__PAGE_OFFSET`" olmadan erişebiliriz. Çünkü çekirdek alanı



(>=3G doğrusal adresinde bulunur) sayfalama etkinleştirildikten sonra doğru bir şekilde kendi fiziksel adresine eşlenecektir.

"lss stack\_start,%esp" (SS:ESP = \*stack\_start) yeni bir yığıt kuran "-PAGE\_OFFSET" olmadan bir sembole başvuran ilk örnektir. BSP için, yığıt `init_task_union`'ın sonundadır. AP için, `stack_start.esp` `linux/arch/i386/kernel/smpboot.c:do_boot_cpu()` tarafından `smp_init()` (sayfa: 53) içinde "(void \*) (1024 + PAGE\_SIZE + (char \*)idle)" olacak şekilde yeniden tanımlanmıştır.

Sayfalama mekanizmaları ve veri yapıları için bakınız: IA-32 Manual Vol.3. (Ch.3.7. Page Translation Using 32-Bit Physical Addressing, Ch.9.8.3. Initializing Paging, Ch.9.9.1. Switching to Protected Mode ve Ch.18.26.3. Enabling and Disabling Paging).

## 6.2. Çekirdek Parametrelerini Al

```
#define OLD_CL_MAGIC_ADDR      0x90020
#define OLD_CL_MAGIC           0xA33F
#define OLD_CL_BASE_ADDR      0x90000
#define OLD_CL_OFFSET         0x90022
#define NEW_CL_POINTER         0x228    /* Gerçek kip veriye göreli */

#ifdef CONFIG_SMP
    if (BX) {
        EFLAGS = 0;                // AP EFLAGS'leri temizler
    } else
#endif
{
    // İlk CPU BSS'yi temizler
    clear BSS;                    // örn. __bss_start .. _end
    setup_idt() {
        /* idt_table[256]; arch/i386/kernel/traps.c içinde tanımlı
         * .data.idt bölümüne yerleştirilmiş
         EAX = __KERNEL_CS << 16 + ignore_int;
         DX = 0x8E00;    // kesme kapısı, dpl = 0, mevcut
         idt_table[0..255] = {EAX, EDX};
        }
        EFLAGS = 0;
        /*
         * Önyükleme parametrelerini yolun dışına kopyala (ayak altından al).
         * _empty_zero_page'in ilk 2kB'lık bölümü önyükleme parametreleri için,
         * ikinci 2kB'lık bölümü komut satırı içindir.
         */
        taşı *ESI (gerçek kip başlık)'dan empty_zero_page'e, 2KB;
        temizle empty_zero_page+2K, 2KB;
        ESI = empty_zero_page[NEW_CL_POINTER];
        if (!ESI) {                // 32 bitlik komut satırı göstericisi
            if (OLD_CL_MAGIC==(uint16)[OLD_CL_MAGIC_ADDR]) {
                ESI = [OLD_CL_BASE_ADDR
                        + (uint16)[OLD_CL_OFFSET];
                taşı *ESI'dan empty_zero_page+2K'ya, 2KB;
            }
        } else {                  // 2.02+'da geçerli
            taşı *ESI'dan empty_zero_page'e, 2KB;
        }
    }
}
```

BSP için çekirdek parametreleri `ESI` tarafından gösterilen bellekten `empty_zero_page`'e kopyalanır. Eğer uygulanabilir ise çekirdek komut satırı `empty_zero_page+2K`'ya kopyalanacaktır.

### 6.3. İşlemci Türünü Kontrol Et

İşlemci türünün ve işlemci içeriğinin nasıl belirlendiği konusunda bakınız: IA-32 Manual Vol.1. (Ch.13. Processor Identification and Feature Determination).

```
struct cpuinfo_x86;           // bakınız: include/asm-i386/processor.h
struct cpuinfo_x86 boot_cpu_data; // bakınız: arch/i386/kernel/setup.c

#define CPU_PARAMS            SYMBOL_NAME(boot_cpu_data)
#define X86                   CPU_PARAMS+0
#define X86_VENDOR           CPU_PARAMS+1
#define X86_MODEL            CPU_PARAMS+2
#define X86_MASK              CPU_PARAMS+3
#define X86_HARD_MATH        CPU_PARAMS+6
#define X86_CPUID             CPU_PARAMS+8
#define X86_CAPABILITY        CPU_PARAMS+12
#define X86_VENDOR_ID        CPU_PARAMS+28

checkCPUtype:
{
    X86_CPUID = -1;           // CPUID yok

    X86 = 3;                  // en azından 386
    save original EFLAGS to ECX;
    flip AC bit (0x40000) in EFLAGS;
    if (AC bit not changed) goto is386;

    X86 = 4;                  // en azından 486
    flip ID bit (0x200000) in EFLAGS;
    restore original EFLAGS;   // AC ve ID seçenekleri için
    if (ID bit değişmez) goto is486;

    // işlemci bilgilerini al
    CPUID(EAX=0);
    X86_CPUID = EAX;
    X86_VENDOR_ID = {EBX, EDX, ECX};
    if (!EAX) goto is486;

    CPUID(EAX=1);
    CL = AL;
    X86 = AH & 0x0f;           // aile
    X86_MODEL = (AL & 0xf0) >> 4; // model
    X86_MASK = CL & 0x0f;       // adımlama kimliği (stepping id)
    X86_CAPABILITY = EDX;       // özellik
}
```

x87 aritmetik işlemcisinin ayarlanması için bakınız: IA-32 Manual Vol.3. (Ch.9.2. x87 FPU Initialization, and Ch.18.14. x87 FPU).

```
is486:
    // PG, PE, ET'yi kaydet AM, WP, NE, MP'yi ise ayarla
    EAX = (CR0 & 0x80000011) | 0x50022;
    goto 2f;                // "is386:" işlemeyi atla
is386:
    orjinal EFLAGS'ları ECX'den yeniden al;
```

```
// PG, PE, ET'yi kaydet MP'yi ayarla
EAX = (CR0 & 0x80000011) | 0x02;

/* ET: Eklenti Türü (Extension Type) (CR0'ın 4 biti).
 * Intel 386 ve Intel 486 işlemcilerde bu seçenek atandığı (set) zaman
 * Intel 387 DX aritmetik işlemcisi komutlarının desteklendiğini gösterir.
 * Pentium 4, Intel Xeon ve P6 ailesi işlemcilerde ise
 * bu seçenek sabit 1 olur.
 * -- IA-32 Manual Vol.3. Ch.2.5. Control Registers (p.2-14) */

2:
CR0 = EAX;
check_x87() {
    /* Doğru olması için ET'ye bağımlıyız.
     * Bu 287/387 için sınar. */
    X86_HARD_MATH = 0;
    clts;                // CR0.TS = 0;
    fninit;              // Init FPU;
    fstsw AX;            // AX = ST(0);
    if (AL) {
        CR0 ^= 0x04;    // yardımcı işlemci (coprocessor) yok, EM'i ata
    } else {
        ALIGN
1:    X86_HARD_MATH = 1;
        /* IA-32 Manual Vol.3. Ch.18.14.7.14. FSETPM komutu
         * 287 işlemcisinin korumalı kipte olduğunu söyler
         * 387 tarafından dikkate alınmaz*/
        fsetpm;
    }
}
}
```

linux/include/linux/linkage.h içinde tanımlanan ALIGN makrosu, 16-byte hizalama ve 0x90 değeri doldurma (NOP için opcode) belirtir. .align talimatının anlamı için ayrıca [Using as: Assembler Directives](#)<sup>(B107)</sup> belgesine bakınız.

## 6.4. Çekirdeği Başlat

```
ready: .byte 0;        // global değişken
{
    ready++;            // kaç işlemci hazır
    lgdt gdt_descr;     // yeni betimleme tablosunu güvenli yerde kullan
    lidt idt_descr;
    goto __KERNEL_CS:$1f; // "lgdt"den sonra bölüt yazmaçlarını yeniden yükle
1:
    DS = ES = FS = GS = __KERNEL_DS;
#ifdef CONFIG_SMP
    SS = __KERNEL_DS;    // sadece bölütü yeniden yükle
#else
    SS:ESP = *stack_start; /* init_task_union'ın sonu,
                           * linux/arch/i386/kernel/init_task.c'de tanımlı */
#endif
    EAX = 0;
    lldt AX;
    cld;
```

```

#ifdef CONFIG_SMP
    if (1!=ready) {          // ilk işlemci değil
        initialize_secondary();
        // bakınız: linux/arch/i386/kernel/smpboot.c
    } else
#endif
    {
        start_kernel(); // bakınız: linux/init/main.c
    }
L6:
    goto L6;
}

```

İlk işlemci (BSP) `linux/init/main.c:start_kernel()`'i çağırarak ve diğerleri (AP) `linux/arch/i386/kernel/smpboot.c:initialize_secondary()`'yi çağırarak. [linux/init/main.c](#) (sayfa: 46) içinde `start_kernel()`'e ve [initialize\\_secondary\(\)](#) (sayfa: 57) içindeki `initialize_secondary()`'e bakınız.

`init_task_union` ilk süreç olan "idle" süreci (pid=0) için görev yapısı olmak üzere oluşur. "idle" sürecinin yığılı `init_task_union`'ın sonundan itibaren gelişir. Aşağıdaki kod `init_task_union` ile ilgilidir.

```

ENTRY(stack_start)
    .long init_task_union+8192;
    .long __KERNEL_DS;

#ifdef INIT_TASK_SIZE
# define INIT_TASK_SIZE 2048*sizeof(long)
#endif

union task_union {
    struct task_struct task;
    unsigned long stack[INIT_TASK_SIZE/sizeof(long)];
};

/* INIT_TASK ilk görev tablosunu kurmak için kullanıldı,
 * riski göze alarak kullanın! Base=0, limit=0x1fffff (=2MB) */
union task_union init_task_union
    __attribute__((__section__(".data.init_task"))) =
    { INIT_TASK(init_task_union.task) };

```

`init_task_union` BSP "idle" süreci içindir. [init\(\)](#) (sayfa: 49) bölümünde değinilen "init" süreciyle karıştırmayın.

## 6.5. Muhtelif

```

////////////////////////////////////
// öntanımlı kesme yöneticisi ("handler")
ignore_int() { printk("Unknown interrupt\n"); iret; }

/*
 * Kesme belirtici tablosu 256 idt için odaya (room) sahiptir,
 * global belirtici tablosu sahip olabileceğimiz
 * görev sayısına bağlıdır...
 */
#define IDT_ENTRIES      256
#define GDT_ENTRIES      ( __TSS(NR_CPUS) )

.globl SYMBOL_NAME(idt)

```

```

.globl SYMBOL_NAME(gdt)

    ALIGN
    .word 0
idt_descr:
    .word IDT_ENTRIES*8-1          # idt 256 girdi içerir
SYMBOL_NAME(idt):
    .long SYMBOL_NAME(idt_table)

    .word 0
gdt_descr:
    .word GDT_ENTRIES*8-1
SYMBOL_NAME(gdt):
    .long SYMBOL_NAME(gdt_table)

/*
 * Bu, 0-8M'de (önyükleme amaçları için) bir kimlik eşleşmesi ve
 * PAGE_OFFSET sanal adresinde başka bir 0-8M eşleşmesi oluşturmak
 * üzere ilklendirilir.
 */
.org 0x1000
ENTRY(swapper_pg_dir)    // "ENTRY" linux/include/linux/linkage.h'da tanımlı
    .long 0x00102007
    .long 0x00103007
    .fill BOOT_USER_PGD_PTRS-2,4,0
/* öntanımlı: 766 girdi */
    .long 0x00102007
    .long 0x00103007
/* öntanımlı: 254 girdi */
    .fill BOOT_KERNEL_PGD_PTRS-2,4,0

/*
 * Sayfa tablolarının burada sadece 8MB'ı ilklendirilir
 * - sonuncu sayfa tabloları bellek boyutuna bağlı
 * olarak daha sonra ayarlanır.
 */
.org 0x2000
ENTRY(pg0)

.org 0x3000
ENTRY(pg1)

/*
 * empty_zero_page hemen sayfa tablosunu takip etmelidir!
 * (İklendirme döngüsü empty_zero_page'e kadar sayar)
 */
.org 0x4000
ENTRY(empty_zero_page)

/*
 * normal "text" bölütünün gerçek başlangıcı
 */
.org 0x5000
ENTRY(stext)
ENTRY(_stext)

////////////////////////////////////

```

```

/*
 * Bu veri bölümünü başlatır. Dikkat ederseniz yukarıda tümü
 * text bölümündedir çünkü bu bizim başka bir şekilde
 * gideremeyeceğimiz hizalama gereksinimidir.
 */
.data

ALIGN
/*
 * Tipik olarak 140 "quadwords" içerir; NR_CPUS'a bağlı olarak.
 *
 * DİKKAT! Herhangi bir şeyi değiştirirseniz, bunun head.S'deki
 * gdt belirticisiyle eşleştigiinden emin olun.
 */
ENTRY(gdt_table)
    .quad 0x0000000000000000    /* NULL belirtici */
    .quad 0x0000000000000000    /* kullanılmadı */
    .quad 0x00cf9a000000ffff    /* 0x10 kernel 4GB code at 0x00000000 */
    .quad 0x00cf92000000ffff    /* 0x18 kernel 4GB data at 0x00000000 */
    .quad 0x00cffa000000ffff    /* 0x23 user 4GB code at 0x00000000 */
    .quad 0x00cff2000000ffff    /* 0x2b user 4GB data at 0x00000000 */
    .quad 0x0000000000000000    /* kullanılmadı */
    .quad 0x0000000000000000    /* kullanılmadı */
/*
 * APM bölütleri bayt taneciklilik özelliğine sahiptir ve
 * tabanları ile sınırları çalışma zamanında atanır.
 */
    .quad 0x0040920000000000    /* 0x40 kötü BIOS'lar için APM ataması */
    .quad 0x00409a0000000000    /* 0x48 APM CS kod */
    .quad 0x00009a0000000000    /* 0x50 APM CS 16 kod (16 bit) */
    .quad 0x0040920000000000    /* 0x58 APM DS veri */
    .fill NR_CPUS*4,8,0        /* TSS'ler ve LDT'ler için boşluk */

```

`idt_descr` ve `gdt_table`'dan önce olan `ALIGN` makrosu performans ile ilgilidir.

## 6.6. Kaynakça

- [IA-32 Intel Architecture Software Developer's Manual<sup>\(B111\)</sup>](#)
- [MultiProcessor Specification<sup>\(B112\)</sup>](#)
- [Using `as`<sup>\(B113\)</sup>](#)
- [GNU Binary Utilities<sup>\(B114\)</sup>](#)

## 7. linux/init/main.c

Bu bölümü yazarken kendimi suçlu hissettim çünkü yeterince olmasa bile hakkında çok sayıda belge var. `start_kernel()` destekli işlevler, sürekli gelişen işletim sistemi dahili bileşenlerine bağımlı olduğu için, sürümden sürüme değişir. Sık sık belge güncellemek için vaktim olmadığından bu bölümü olabildiğince basit tutmaya karar verdim.

### 7.1. start\_kernel()

```

////////////////////////////////////
asmlinkage(B115) void __init(B116) start_kernel(void)

```

```

{
    char * command_line;
    extern char saved_command_line[];
/*
 * Kesmeler hala etkin değil. Gerekli kurulumu yap, sonra etkinleştir
 */
    lock_kernel();
    printk(linux_banner);

    /* Linux'da bellek yönetimi(B117), esp. for setup_arch()
     * Linux-2.4.4 MM Başlangıç durumuna getirme(B118) */
    setup_arch(&command_line);
    printk("Kernel command line: %s\n", saved_command_line);

    /* linux/Documentation/kernel-parameters.txt
     * The Linux BootPrompt-HowTo(B119) */
    parse_options(command_line);

    trap_init() {
#ifdef CONFIG_EISA
        if (isa_readl(0x0FFFD9) == 'E' + ('I' << 8) + ('S' << 16) + ('A' << 24))
            EISA_bus = 1;
#endif
#ifdef CONFIG_X86_LOCAL_APIC
        init_apic_mappings();
#endif
        set_xxxx_gate(x, &func);    // kurulum kapıları
        cpu_init();
    }
    init_IRQ();
    sched_init();
    softirq_init() {
        for (int i=0; i<32; i++)
            tasklet_init(bh_task_vec+i, bh_action, i);
        open_softirq(TASKLET_SOFTIRQ, tasklet_action, NULL);
        open_softirq(HI_SOFTIRQ, tasklet_hi_action, NULL);
    }
    time_init();

/*
 * HACK ALERT! Bu erken. PCI ve bunun gibi kurulumları bitirdikten
 * ve console_init()'in bunu farketmesinde önce konsolu etkinleştirmeliyiz.
 * Birşeylerin kötü gitmesi durumunda bunun erkenden çıktığı olmasını isteriz.
 */
    console_init();
#ifdef CONFIG_MODULES
    init_modules();
#endif
    if (prof_shift) {
        unsigned int size;
        /* only text is profiled */
        prof_len = (unsigned long) &_etext - (unsigned long) &_stext;
        prof_len >>= prof_shift;
        size = prof_len * sizeof(unsigned int) + PAGE_SIZE-1;
        prof_buffer = (unsigned int *) alloc_bootmem(size);
    }
}

```

```

kmem_cache_init();
sti();

// Bogomips mini-Howto(B120)
calibrate_delay();

// linux/Documentation/initrd.txt
#ifdef CONFIG_BLK_DEV_INITRD
    if (initrd_start && !initrd_below_start_ok &&
        initrd_start < min_low_pfn << PAGE_SHIFT) {
        printk(KERN_CRIT "initrd overwritten (0x%08lx < 0x%08lx) - "
            "disabling it.\n", initrd_start, min_low_pfn << PAGE_SHIFT);
        initrd_start = 0;
    }
#endif

mem_init();
kmem_cache_sizes_init();
pgtable_cache_init();

/*
 * Yüksek belleğe (highmem) sahip olan mimariler için, num_mappedpages
 * çekirdeğin kullanabileceği bellek miktarını ifade eder. Diğer mimariler
 * için toplam sayfa ile aynıdır. Her iki rakama da ihtiyaç duyarız çünkü
 * bazı altsistemler çekirdeğin ne kadar bellek kullanabileceğine dayanarak
 * başlangıç durumuna getirilir.
 */
if (num_mappedpages == 0)
    num_mappedpages = num_physpages;

fork_init(num_mempages);
proc_caches_init();
vfs_caches_init(num_physpages);
buffer_init(num_physpages);
page_cache_init(num_physpages);
#ifdef CONFIG_ARCH_S390
    ccwcache_init();
#endif
signals_init();
#ifdef CONFIG_PROC_FS
    proc_root_init();
#endif
#ifdef CONFIG_SYSVIPC
    ipc_init();
#endif
check_bugs();
printk("POSIX conformance testing by UNIFIX\n");

/*
 * İlk işlemdeki (thread) iyi gidenleri sayarız
 * atıl (idlers) gibi init de kilitsiz bir çekirdek işlemidir,
 * sistem çağrısı yapar (ve böylece kilitlenir).
 */
smp_init() {
#ifdef CONFIG_SMP
#ifdef CONFIG_X86_LOCAL_APIC
    APIC_init_uniprocessor();

```



```
# else
    do { } while (0);
# endif
#else
    /* Check smp_init() (sayfa: 53). */
#endif
}

rest_init() {
    // init process, pid = 1
    kernel_thread(init, NULL, CLONE_FS | CLONE_FILES | CLONE_SIGNAL);
    unlock_kernel();
    current->need_resched = 1;
    // idle process, pid = 0
    cpu_idle();      // never return
}
}
```

`start_kernel()` "init" işlemi oluşturmak için `rest_init()` 'i çağırır ve kendisi "idle" işlem durumuna geçer.

## 7.2. init()

"Init" süreci:

```
////////////////////////////////////
static int init(void * unused)
{
    lock_kernel();
    do_basic_setup();

    prepare_namespace();

    /*
     * Tamam, ilk önyüklemeyi (bootup) tamamladık, ayakta ve
     * çalışır durumdayız. initmem bölütlerinden kurtul ve
     * kullanıcı kipini başlat...
     */
    free_initmem();
    unlock_kernel();

    if (open("/dev/console", O_RDWR, 0) < 0)      // stdin
        printk("Warning: unable to open an initial console.\n");

    (void) dup(0);                                // stdout
    (void) dup(0);                                // stderr

    /*
     * Biri başarılı olana kadar her birini deneyeceğiz
     * Gerçekten bozuk bir makineyi toparlamaya çalışıyorsak
     * init yerine Bourne kabuğu kullanılabilir.
     */

    if (execute_command)
        execve(execute_command, argv_init, envp_init);
    execve("/sbin/init", argv_init, envp_init);
    execve("/etc/init", argv_init, envp_init);
    execve("/bin/init", argv_init, envp_init);
}
```

```

execve("/bin/sh",argv_init,envp_init);
panic("No init found. Try passing init= option to kernel.");
}

```

Kullanıcı kipi "init" süreciyle ilgili bilgiler için **man init** veya **SysVinit**<sup>(B122)</sup>'e bakınız.

### 7.3. cpu\_idle()

"Idle" süreç:

```

/*
 * Boşta bekleme (idle) evresi. Yapılacak yararlı bir iş yok,
 * bu yüzden sadece gücü korumaya çalış ve düşük çıkış gecikmesine
 * sahip ol (örn. birinin yeniden iş yapma isteği
 * belirtmesini bekleyen bir döngü içinde kal)
 */
void cpu_idle (void)
{
    /* hiç bir önceliği olmayan sonsuz atıl döngü */
    init_idle();
    current->nice = 20;
    current->counter = -100;

    while (1) {
        void (*idle)(void) = pm_idle;
        if (!idle)
            idle = default_idle;
        while (!current->need_resched)
            idle();
        schedule();
        check_pgt_cache();
    }
}

/////////////////////////////////////////////////////////////////
void __init init_idle(void)
{
    struct schedule_data * sched_data;
    sched_data = &aligned_data[smp_processor_id()].schedule_data;

    if (current != &init_task && task_on_runqueue(current)) {
        printk("UGH! (%d:%d) was on the runqueue, removing.\n",
            smp_processor_id(), current->pid);
        del_from_runqueue(current);
    }
    sched_data->curr = current;
    sched_data->last_schedule = get_cycles();
    clear_bit(current->processor, &wait_init_idle);
}

/////////////////////////////////////////////////////////////////
void default_idle(void)
{
    if (current_cpu_data.hlt_works_ok && !hlt_counter) {
        __cli();
        if (!current->need_resched)
            safe_halt();
    }
}

```

```

    else
        __sti();
    }
}

/* linux/include/asm-i386/system.h içinde tanımlı */
#define __cli()          __asm__ __volatile__ ("cli":::"memory")
#define __sti()          __asm__ __volatile__ ("sti":::"memory")

/* atıl döngü içinde kullanıldı; sti'nin tamamlanması bir komut süresi alır */
#define safe_halt()      __asm__ __volatile__ ("sti; hlt":::"memory")

```

İşlemci bir kesme eylemcisinden dönen "hlt"yi takip eden komut ile kod çalıştırmaya devam edecektir.

## 7.4. Kaynakça

- [Linux Kernel 2.4 Internals](#)<sup>(B123)</sup>
- [Kerneldoc](#)<sup>(B124)</sup>
- [LDP HOWTO-INDEX](#)<sup>(B125)</sup>
- [Linux Device Drivers, 2nd Edition](#)<sup>(B126)</sup>

## 8. SMP Önyükeme

SMP ile ilgili `CONFIG_SMP`, `CONFIG_X86_LOCAL_APIC`, `CONFIG_X86_IO_APIC`, `CONFIG_MULTIQUEAD` ve `CONFIG_VISWS` gibi bir kaç makro bulunmaktadır. Birçok insanın dikkate almadığı (eğer IBM üst-uç çok işlemcili sunucu veya SGI Görsel İşistasyonu kullanmıyorsa) `CONFIG_MULTIQUEAD` veya `CONFIG_VISWS` gerektiren kodları yoksayacağım.

BSP APLeri tetiklemek için `start_kernel()` -> `smp_init()` -> `smp_boot_cpus()` -> `do_boot_cpu()` -> `wakeup_secondary_via_INIT()`'i çalıştırır. Teknik ayrıntılar için bakınız: [MultiProcessor Specification](#)<sup>(B127)</sup> ve IA-32 Manual Vol.3 (Ch.7. Multile-Processor Management, and Ch.8. Advanced Programmable Interrupt Controller).

### 8.1. smp\_init() Öncesi

`smp_init()`'i çağırmadan önce, `start_kernel()` SMP ortamını kurmak için birşeyler yapar.

```

start_kernel()
|-- setup_arch()
|   |-- parse_cmdline_early(); // SMP "noht" ve "acpism=force" arar
|   |   |-- /* "noht" çoklu işlem (HyperThreading) özelliğini iptal eder
|   |   |   * (Xeon başına 2 mantıksal işlemci) */
|   |   |   if (!memcmp(from, "noht", 4)) {
|   |   |       disable_x86_ht = 1;
|   |   |       set_bit(X86_FEATURE_HT, disabled_x86_caps);
|   |   |   }
|   |   |   /* "acpism=force" ACPI SMP tablosunu kullanıma ve
|   |   |   * çözümlemeye zorlar */
|   |   |   else if (!memcmp(from, "acpism=force", 13))
|   |   |       enable_acpi_smp_table = 1;
|   |-- setup_memory(); // MP yapılandırma tablosu için bellek ayırır
|   |   |-- reserve_bootmem(PAGE_SIZE, PAGE_SIZE);
|   |   |-- find_smp_config();
|   |   |-- find_intel_smp();

```

```

|         |-- smp_scan_config();
|         |         |-- set flag smp_found_config
|         |         |-- set MP floating pointer mpf_found
|         |         |-- reserve_bootmem(mpf_found, PAGE_SIZE);
| |-- if (disable_x86_ht) { // çoklu işlem iptal edilmişse
| |     clear_bit(X86_FEATURE_HT, &boot_cpu_data.x86_capability[0]);
| |     set_bit(X86_FEATURE_HT, disabled_x86_caps);
| |     enable_acpi_smp_table = 0;
| | }
| |-- if (test_bit(X86_FEATURE_HT, &boot_cpu_data.x86_capability[0]))
| |     enable_acpi_smp_table = 1;
| |-- smp_alloc_memory();
| |     |-- /* AP işlemcilerinin gerçek kip kod uzayını düşük bellekte ayırır */
| |     trampoline_base = (void *) alloc_bootmem_low_pages(PAGE_SIZE);
| |-- get_smp_config(); // Önyükeme anı MP yapılandırmasını al */
| |     |-- config_acpi_tables();
| | |         |-- memset(&acpi_boot_ops, 0, sizeof(acpi_boot_ops));
| | |         |-- acpi_boot_ops[ACPI_APIC] = acpi_parse_madt;
| | |         |-- /* have_acpi_tables değerini ACPI tablosunda
| | |         * MADT kullanıldığını belirtecek şekilde değiştirir;
| | |         * başarısız olursa MPS kullan. */
| | |         if (enable_acpi_smp_table && !acpi_tables_init())
| | |             have_acpi_tables = 1;
| | |-- set pic_mode
| | |     /* =1, IMCR varsa ve PIC Kipi gerçekleştirilmişse;
| | |     * =0, aksi taktirde Sanal Tel Tipi (Virtual Wire Mode)
| | |     * gerçekleştirilmiştir. */
| | |-- mp_lapic_addr'deki yerel APIC adresini sakla
| | |-- MP_PROCESSOR, MP_BUS, MP_IOAPIC, MP_INTSRC ve MP_LINTSRC.MP gibi
| |     yapılandırma tablo girdilerini tara.
|
|-- trap_init();
| |-- init_apic_mappings(); // PTE'yi APIC için ayarla
| |     |-- /* Eğer hiç yerel APIC bulunmazsa tüm sıfır sayfalarını
| |     * (zeroes page) yerel APIC'i ve bir de IO-APIC'i
| |     * taklit edecek şekilde ayarla. */
| |     if (!smp_found_config && detect_init_APIC()) {
| |         apic_phys = (unsigned long) alloc_bootmem_pages(PAGE_SIZE);
| |         apic_phys = __pa(apic_phys);
| |     } else
| |         apic_phys = mp_lapic_addr;
| |-- /* yerel APIC adresini,
| |     * çoğu durumda mp_lapic_addr'dir (0xfe00000),
| |     * bunu doğrusal FIXADDR_TOP (0xffffe000) adresine eşle */
| |     set_fixmap_nocache(FIX_APIC_BASE, apic_phys);
| |-- /* Öntanımlı yapılandırmamız (veya MP tablosunun bozuk olması)
| |     * durumu için BSP'nin APIC ID değerini al. */
| |     if (boot_cpu_physical_apicid == -1U)
| |         boot_cpu_physical_apicid = GET_APIC_ID(apic_read(APIC_ID));
| |-- /* IOAPIC adresini arabelleksiz doğrusal adrese eşle
| |     set_fixmap_nocache(idx, ioapic_phys);
| |     // Artık APIC uzayına erişmek için doğrusal adres kullanabiliriz.
|-- init_IRQ();
| |-- init_ISA_irqs();
| |     |-- /* sanal tel kipinin ilk kurulumu. */
| |     init_bsp_APIC();
| |     |-- init_8259A(auto_eoi=0);

```

```
|  '-- setup SMP/APIC interrupt handlers, esp. IPI.
|-- mem_init();
    '-- /* SMP için düşük adresleme geçişini (zapping)
        * geciktir: zap_low_mappings() */
```

IPI (İşlemcilerarası Kesme – InterProcessor Interrupt), yani işlemciden işlemciye yerel APIC üzerinden kesme, BSP'nin APLeri tetiklemek için kullandığı mekanizmadır.

Bilmeniz gerekir ki MP-uyumlu sistemlerde "CPU başına bir yerel APIC gereklidir". İşlemciler APIC yerel birimlerinin adres uzayını (0xFEE00000 – 0xFEEFFFFFF fiziksel adresini) paylaşmazlar, fakat APIC giriş-çıkış birimlerini (0xFEC00000 – 0xFECFFFFFF) paylaşırlar. Her iki adres uzayı da arabelleklenebilir değildir.

## 8.2. smp\_init()

BSP her bir işlemcinin veri yapılarını kurmak ve kalan APLeri etkinleştirmek için `start_kernel()` -> `smp_init()` -> `smp_boot_cpus()` çağrısı yapar.

```
////////////////////////////////////
static void __init smp_init(void)
{
    /* Get other processors into their bootup holding patterns. */
    smp_boot_cpus();
    wait_init_idle = cpu_online_map;
    clear_bit(current->processor, &wait_init_idle); /* Don't wait on me! */

    smp_threads_ready=1;
    smp_commence() {
        /* Lets the callins below out of their loop. */
        Dprintk("Setting commenced=1, go go go\n");
        wmb();
        atomic_set(&smp_commenced,1);
    }

    /* Wait for the other cpus to set up their idle processes */
    printk("Waiting on wait_init_idle (map = 0x%lx)\n", wait_init_idle);
    while (wait_init_idle) {
        cpu_relax(); // i.e. "rep;nop"
        barrier();
    }
    printk("All processors have done init_idle\n");
}

////////////////////////////////////
void __init smp_boot_cpus(void)
{
    // ... çok ilginç olmayan birşey:-)

    /* Initialize the logical to physical CPU number mapping
     * and the per-CPU profiling router/multiplier */
    prof_counter[0..NR_CPUS-1] = 0;
    prof_old_multiplier[0..NR_CPUS-1] = 0;
    prof_multiplier[0..NR_CPUS-1] = 0;

    init_cpu_to_apicid() {
        physical_apicid_2_cpu[0..MAX_APICID-1] = -1;
        logical_apicid_2_cpu[0..MAX_APICID-1] = -1;
        cpu_2_physical_apicid[0..NR_CPUS-1] = 0;
```

```
    cpu_2_logical_apicid[0..NR_CPUS-1] = 0;
}

/* Setup boot CPU information */
smp_store_cpu_info(0); /* Final full version of the data */
printk("CPU%d: ", 0);
print_cpu_info(&cpu_data[0]);

/* We have the boot CPU online for sure. */
set_bit(0, &cpu_online_map);
boot_cpu_logical_apicid = logical_smp_processor_id() {
    GET_APIC_LOGICAL_ID(*(unsigned long *) (APIC_BASE+APIC_LDR));
}
map_cpu_to_boot_apicid(0, boot_cpu_apicid) {
    physical_apicid_2_cpu[boot_cpu_apicid] = 0;
    cpu_2_physical_apicid[0] = boot_cpu_apicid;
}

global_irq_holder = 0;
current->processor = 0;
init_idle();    // will clear corresponding bit in wait_init_idle
smp_tune_scheduling();

// ... some conditions checked

connect_bsp_APIC();    // enable APIC mode if used to be PIC mode
setup_local_APIC();

if (GET_APIC_ID(apic_read(APIC_ID)) != boot_cpu_physical_apicid)
    BUG();

/* Scan the CPU present map and fire up the other CPUs
 *   via do_boot_cpu() */
Dprintk("CPU present map: %lx\n", phys_cpu_present_map);
for (bit = 0; bit < NR_CPUS; bit++) {
    apicid = cpu_present_to_apicid(bit);
    /* Don't even attempt to start the boot CPU! */
    if (apicid == boot_cpu_apicid)
        continue;
    if (!(phys_cpu_present_map & (1 << bit)))
        continue;
    if ((max_cpus >= 0) && (max_cpus <= cpucount+1))
        continue;
    do_boot_cpu(apicid);
    /* Make sure we unmap all failed CPUs */
    if ((boot_apicid_to_cpu(apicid) == -1) &&
        (phys_cpu_present_map & (1 << bit)))
        printk("CPU #%d not responding - cannot use it.\n", apicid);
}

// ... SMP BogoMIPS
// ... B stepping processor warning
// ... HyperThreading handling

/* Set up all local APIC timers in the system */
setup_APIC_clocks();
```

```

/* Synchronize the TSC with the AP */
if (cpu_has_tsc && cpucount)
    synchronize_tsc_bp();

smp_done:
    zap_low_mappings();
}

////////////////////////////////////
static void __init do_boot_cpu (int apicid)
{
    cpu = ++cpucount;

    // 1. prepare "idle process" task struct for next AP

    /* We can't use kernel_thread since we must avoid to
     * reschedule the child. */
    if (fork_by_hand() < 0)
        panic("failed fork for CPU %d", cpu);
    /* We remove it from the pidhash and the runqueue
     * once we got the process: */
    idle = init_task.prev_task;
    if (!idle)
        panic("No idle process for CPU %d", cpu);

    /* we schedule the first task manually */
    idle->processor = cpu;
    idle->cpus_runnable = 1 << cpu; // only on this AP!

    map_cpu_to_boot_apicid(cpu, apicid) {
        physical_apicid_2_cpu[apicid] = cpu;
        cpu_2_physical_apicid[cpu] = apicid;
    }

    idle->thread.eip = (unsigned long) start_secondary;

    del_from_runqueue(idle);
    unhash_process(idle);
    init_tasks[cpu] = idle;

    // 2. prepare stack and code (CS:IP) for next AP

    /* start_eip had better be page-aligned! */
    start_eip = setup_trampoline() {
        memcpy(trampoline_base, trampoline_data,
               trampoline_end - trampoline_data);
        /* trampoline_base was reserved in
         * start_kernel() -> setup_arch() -> smp_alloc_memory(),
         * and will be shared by all APs (one by one) */
        return virt_to_phys(trampoline_base);
    }

    /* So we see what's up */
    printk("Booting processor %d/%d eip %lx\n", cpu, apicid, start_eip);
    stack_start.esp = (void *) (1024 + PAGE_SIZE + (char *)idle);
    /* this value is used by next AP when it executes
     * "lss stack_start,%esp" in

```

```

*   linux/arch/i386/kernel/head.S:startup_32(). */

/* This grunge runs the startup process for
 * the targeted processor. */
atomic_set(&init_deasserted, 0);
Dprintk("Setting warm reset code and vector.\n");

CMOS_WRITE(0xa, 0xf);
local_flush_tlb();
Dprintk("1.\n");
*((volatile unsigned short *) TRAMPOLINE_HIGH) = start_eip >> 4;
Dprintk("2.\n");
*((volatile unsigned short *) TRAMPOLINE_LOW) = start_eip & 0xf;
Dprintk("3.\n");
// we have setup 0:467 to start_eip (trampoline_base)

// 3. kick AP to run (AP gets CS:IP from 0:467)

// Starting actual IPI sequence...
boot_error = wakeup_secondary_via_INIT(apicid, start_eip);
if (!boot_error) {           // looks OK
    /* allow APs to start initializing. */
    set_bit(cpu, &cpu_callout_map);

    /* ... Wait 5s total for a response */

    // bit cpu in cpu_callin_map is set by AP in smp_callin()
    if (test_bit(cpu, &cpu_callin_map)) {
        print_cpu_info(&cpu_data[cpu]);
    } else {
        boot_error= 1;
        // marker 0xA5 set by AP in trampoline_data()
        if (*((volatile unsigned char *)phys_to_virt(8192)) == 0xA5)
            /* trampoline started but... */
            printk("Stuck ??\n");
        else
            /* trampoline code not run */
            printk("Not responding.\n");
    }
}
if (boot_error) {
    /* Try to put things back the way they were before ... */
    unmap_cpu_to_boot_apicid(cpu, apicid);
    clear_bit(cpu, &cpu_callout_map); /* set in do_boot_cpu() */
    clear_bit(cpu, &cpu_initialized); /* set in cpu_init() */
    clear_bit(cpu, &cpu_online_map); /* set in smp_callin() */
    cpucount--;
}

/* mark "stuck" area as not stuck */
*((volatile unsigned long *)phys_to_virt(8192)) = 0;
}

```

`start_secondary()` ile `trampoline_data()`'yı karıştırmayın. İlki AP "idle" süreci görev yapısı EIP değeridir, ikincisi ise AP'nin BSP onu uyardıktan sonra (`wakeup_secondary_via_INIT()` kullanarak) çalıştırdığı gerçek kipli koddur.



### 8.3. linux/arch/i386/kernel/trampoline.S

Bu dosya 16 bitlik gerçek kipli AP başlatma kodunu içerir. BSP `start_kernel()` -> `setup_arch()` -> `smp_alloc_memory()` içinde `trampoline_base` bellek uzayını ayırdı. BSP APyi tetiklemeden önce `trampoline` kodunu `trampoline_data` ve `trampoline_end` arasından `trampoline_base`'e (`do_boot_cpu()` -> `setup_trampoline()` içindeki) kopyalar. BSP 0:467 noktasını, AP bu noktadan itibaren çalışabilmesi için, `trampoline_base` olarak atar.

```

////////////////////////////////////
trampoline_data()
{
r_base:
    wbinvd;          // diğeri için zararsız olması gereken NUMA-Q için gerekli
    DS = CS;
    BX = 1;          // Bir SMP trampoline seçeneği
    cli;

    // master'in çalıştığımızı bilmesi için imleyiciyi yaz
    trampoline_base = 0xA5A5A5A5;

    lidt idt_48;
    lgdt gdt_48;

    AX = 1;
    lmsw AX;          // korumalı kip!
    goto flush_instr;
flush_instr:
    goto CS:100000; // see linux/arch/i386/kernel/head.S:startup_32()
}

idt_48:
    .word    0                # idt limit = 0
    .word    0, 0            # idt base = 0L

gdt_48:
    .word    0x0800           # gdt limit = 2048, 256 GDT girdisi
    .long    gdt_table-__PAGE_OFFSET # gdt base = gdt (ilk SMP CPU)

.globl SYMBOL_NAME(trampoline_end)
SYMBOL_NAME_LABEL(trampoline_end)

```

Dikkat ederseniz AP `linux/arch/i386/kernel/head.S:startup_32()`'ye geçtiğinde, BSP-ninkinden (BX=0) farklı olarak, BX=1 oluyor. Bakınız: [linux/arch/i386/kernel/head.S](#) (sayfa: 39).

### 8.4. initialize\_secondary()

BSP'den farklı olarak, [Çekirdeği Başlat](#) (sayfa: 43) bölümündeki

`linux/arch/i386/kernel/head.S:startup_32()`'nin sonunda AP `start_kernel()` yerine `initialize_secondary()`'yi çağırır.

```

/* İkincil işlemciler için herşey kurulmuş durumda;
 * bunlar sadece görev yapısındaki herşeyi yeniden
 * yüklemek için gerekli.
 * Bu işlev değer döndürmemeli. */
void __init initialize_secondary(void)
{
    /* Aslında tüm TSS'yi yüklememiz gerekmiyor,

```

```

    * basitçe sadece yığıt göstericisi ve eip. */
asm volatile(
    "movl %0,%%esp\n\t"
    "jmp *%1"
    :
    : "r" (current->thread.esp), "r" (current->thread.eip));
}

```

BSPnin `thread.eip`'yi `start_secondary()` 'ye atamak için `do_boot_cpu()` 'yu çağırmasıyla denetim bu işleve geçer. AP, BSP tarafından `do_boot_cpu()` -> `fork_by_hand()` -> `do_fork()` içinde kurulan, yeni bir yığıt çerçevesi kullanır.

## 8.5. start\_secondary()

Tüm APlar `smp_init()` (sayfa: 53) içindeki `smp_init()` -> `smp_commence()` ile tetiklenen BSP'den `smp_commenced` sinyalini beklerler. Bu sinyali aldıktan sonra "idle" süreçlerini çalıştırır.

```

////////////////////////////////////
int __init start_secondary(void *unused)
{
    /* smp_callin() öncesine birşey koyma, en yapılması gereken şeyleri
     * yapmak için yapılacak şeyleri sınırlamak istediğimiz bu yerde
     * SMP önyüklemesi çok kırılıgandır.*/
    cpu_init();
    smp_callin();
    while (!atomic_read(&smp_commenced))
        rep_nop();
    /* düşük bellek adreslemesi temizlendi,
     * onları yerel TLB'lerden de çıkar (flush). */
    local_flush_tlb();
    return cpu_idle();          // asla dönmez, bakınız: cpu_idle() (sayfa: 50)
}

```

`cpu_idle()` -> `init_idle()` `wait_init_idle` içindeki ilgili bitleri temizler ve son olarak BSPnin `smp_init()`'i bitirmesini sağlar ve `start_kernel()` içindeki takip eden işlemlerle devam eder (örn. `rest_init()`).

## 8.6. Kaynakça

- [MultiProcessor Specification<sup>\(B132\)</sup>](#)
- [IA-32 Intel Architecture Software Developer's Manual<sup>\(B133\)</sup>](#)
- [Linux Kernel 2.4 Internals: Ch.1.7. SMP Bootup on x86<sup>\(B134\)</sup>](#)
- [Linux SMP HOWTO<sup>\(B135\)</sup>](#)
- [ACPI spec<sup>\(B136\)</sup>](#)
- Bir çokişlemcili Linux gerçeklenimi: `linux/Documentation/smp.tex`

## A. Çekirdek Derleme ve Kurulum Örneği

Aşağıda bir çekirdek kurulum örneği var ([Redhat<sup>\(B137\)</sup>](#) 9.0 üzerinde). "/" ve "/" arasındaki ifadeler konsol çıktısı değil yorum satırlarıdır.

```
[root@localhost root]# ln -s /usr/src/linux-2.4.20 /usr/src/linux
[root@localhost root]# cd /usr/src/linux
[root@localhost linux]# make xconfig
/* .config dosyasını oluşturun:
 * 1. "Load Configuration from File" ile
 * /boot/config-2.4.20-28.9 veya istediğiniz bir config
 * dosyasını yükleyin.
 * 2. Çekirdek yapılandırma parametrelerini değiştirin.
 * 3. "Kaydedip çıkın" */
[root@localhost linux]# make oldconfig
/* isterseniz .config'i yeniden kontrol edin */
[root@localhost linux]# vi Makefile
/* isterseniz linux/Makefile içindeki EXTRAVERSION
 * değerini değiştirin */
[root@localhost linux]# make dep
/* .depend vs. yi oluşturun */
[root@localhost linux]# make bzImage
/* ... Bazı çıktılar atlandı */
ld -m elf_i386 -T /usr/src/linux-2.4.20/arch/i386/vmlinux.lds -e stext arch/i386
/kernel/head.o arch/i386/kernel/init_task.o init/main.o init/version.o init/do_m
ounts.o \
    --start-group \
    arch/i386/kernel/kernel.o arch/i386/mm/mm.o kernel/kernel.o mm/mm.o fs/f
s.o ipc/ipc.o \
    drivers/char/char.o drivers/block/block.o drivers/misc/misc.o drivers/n
et/net.o drivers/media/media.o drivers/char/drm/drm.o drivers/net/fc/fc.o driver
s/net/appletalk/appletalk.o drivers/net/tokenring/tr.o drivers/net/wan/wan.o dri
vers/atm/atm.o drivers/ide/idedriver.o drivers/cdrom/driver.o drivers/pci/driver
.o drivers/net/pcmcia/pcmcia_net.o drivers/net/wireless/wireless_net.o drivers/p
np/pnp.o drivers/video/video.o drivers/net/hamradio/hamradio.o drivers/md/mddev.
o drivers/isdn/vmlinux-obj.o \
    net/network.o \
    /usr/src/linux-2.4.20/arch/i386/lib/lib.a /usr/src/linux-2.4.20/lib/lib.
a /usr/src/linux-2.4.20/arch/i386/lib/lib.a \
    --end-group \
    -o vmlinux
nm vmlinux | grep -v '\(compiled\)\|\(\.o$\)\|\( [aUw] \)\|\(\.\.ng$\)\|\(LASH[R
L]DI\)' | sort > System.map
make[1]: Entering directory `/usr/src/linux-2.4.20/arch/i386/boot'
gcc -E -D__KERNEL__ -I/usr/src/linux-2.4.20/include -D__BIG_KERNEL__ -traditional
-D__ASSEMBLY__ -DSVGA_MODE=NORMAL_VGA bootsect.S -o bbootsect.s
as -o bbootsect.o bbootsect.s
bootsect.S: Assembler messages:
bootsect.S:239: Warning: indirect lcall without '*'
ld -m elf_i386 -Ttext 0x0 -s --oformat binary bbootsect.o -o bbootsect
gcc -E -D__KERNEL__ -I/usr/src/linux-2.4.20/include -D__BIG_KERNEL__ -D__ASSEMBL
Y__ -traditional -DSVGA_MODE=NORMAL_VGA setup.S -o bsetup.s
as -o bsetup.o bsetup.s
setup.S: Assembler messages:
setup.S:230: Warning: indirect lcall without '*'
ld -m elf_i386 -Ttext 0x0 -s --oformat binary -e begtext -o bsetup bsetup.o
make[2]: Entering directory `/usr/src/linux-2.4.20/arch/i386/boot/compressed'
```

```

tmpiggy=_tmp_$$piggy; \
rm -f $tmpiggy $tmpiggy.gz $tmpiggy.lnk; \
objcopy -O binary -R .note -R .comment -S /usr/src/linux-2.4.20/vmlinux $tmpiggy; \
gzip -f -9 < $tmpiggy > $tmpiggy.gz; \
echo "SECTIONS { .data: { input_len = .; LONG(input_data_end - input_data) input_data = .; *(.data) input_data_end = .; }}" > $tmpiggy.lnk; \
ld -m elf_i386 -r -o piggy.o -b binary $tmpiggy.gz -b elf32-i386 -T $tmpiggy.lnk; \
rm -f $tmpiggy $tmpiggy.gz $tmpiggy.lnk
gcc -D__ASSEMBLY__ -D__KERNEL__ -I/usr/src/linux-2.4.20/include -traditional -c head.S
gcc -D__KERNEL__ -I/usr/src/linux-2.4.20/include -Wall -Wstrict-prototypes -Wno-trigraphs -O2 -fno-strict-aliasing -fno-common -fomit-frame-pointer -pipe -mpreferred-stack-boundary=2 -march=i686 -DKBUILD_BASENAME=misc -c misc.c
ld -m elf_i386 -Ttext 0x100000 -e startup_32 -o bvmlinux head.o misc.o piggy.o
make[2]: Leaving directory `/usr/src/linux-2.4.20/arch/i386/boot/compressed'
gcc -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer -o tools/build tools/build.c -I/usr/src/linux-2.4.20/include
objcopy -O binary -R .note -R .comment -S compressed/bvmlinux compressed/bvmlinux.out
tools/build -b bbootsect bsetup compressed/bvmlinux.out CURRENT > bzImage
Root device is (3, 67)
Boot sector 512 bytes.
Setup is 4780 bytes.
System is 852 kB
make[1]: Leaving directory `/usr/src/linux-2.4.20/arch/i386/boot'
[root@localhost linux]# make modules modules_install
/* ... Bazı çıktılar atlandı */
cd /lib/modules/2.4.20; \
mkdir -p pcmcia; \
find kernel -path '*/pcmcia/*' -name '*.o' | xargs -i -r ln -sf ../{} pcmcia
if [ -r System.map ]; then /sbin/depmod -ae -F System.map 2.4.20; fi
[root@localhost linux]# cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.20
[root@localhost linux]# cp vmlinux /boot/vmlinux-2.4.20
[root@localhost linux]# cp System.map /boot/System.map-2.4.20
[root@localhost linux]# cp .config /boot/config-2.4.20
[root@localhost linux]# mkinitrd /boot/initrd-2.4.20.img 2.4.20
[root@localhost linux]# vi /boot/grub/grub.conf
/* Aşağıdaki satırları grub.conf dosyasına ekleyin:
title Linux (2.4.20)
kernel /vmlinuz-2.4.20 ro root=LABEL=/
initrd /initrd-2.4.20.img */

```

Daha fazla ayrıntı için [Kernelnewbies FAQ: How do I compile a kernel](#)<sup>(B138)</sup> ve [Kernel Rebuild Procedure](#)<sup>(B139)</sup> belgelerine başvurunuz.

Çekirdeği [Debian](#)<sup>(B140)</sup> üzerinde kurmak için ayrıca [Debian Installation Manual: Compiling a New Kernel](#)<sup>(B141)</sup>, [The Debian GNU/Linux FAQ: Debian and the kernel](#)<sup>(B142)</sup> ve [Debian Reference: The Linux kernel under Debian](#)<sup>(B143)</sup> belgelerine başvurunuz. Eğer bir sorunla karşılaşırsanız "**zless /usr/share/doc/kernel-package/Problems.gz**" dosyasına bakınız.

## B. Dahili İlintileme Betiği

-T (--script=) seçeneği olmadan, hedefleri ilintilemek için **ld** bu yerleşik betiği kullanacaktır:

```
[root@localhost linux]# ld --verbose
GNU ld version 2.13.90.0.18 20030206
Desteklenen öykünümler:
    elf_i386
    i386linux
iç ilintileme betiği kullanılıyor:
=====
/* -z combrelloc için betik: yeniden yerleştirme (reloc) bölümlerini
 * birleştirir ve sıralar */
OUTPUT_FORMAT("elf32-i386", "elf32-i386",
              "elf32-i386")
OUTPUT_ARCH(i386)
ENTRY(_start)
SEARCH_DIR("/usr/i386-redhat-linux/lib"); SEARCH_DIR("/usr/lib"); SEARCH_DIR("/u
sr/local/lib"); SEARCH_DIR("/lib");
/* "elf" için bunlardan herhangi birine ihtiyacımız var mı?
   __DYNAMIC = 0;      */
SECTIONS
{
    /* Salt okunur bölümler, metin bölütü üzerinde birleştirilmiş:*/
    . = 0x08048000 + SIZEOF_HEADERS;
    .interp      : { *(.interp) }
    .hash        : { *(.hash) }
    .dynsym      : { *(.dynsym) }
    .dynstr      : { *(.dynstr) }
    .gnu.version : { *(.gnu.version) }
    .gnu.version_d : { *(.gnu.version_d) }
    .gnu.version_r : { *(.gnu.version_r) }
    .rel.dyn     :
    {
        *(.rel.init)
        *(.rel.text .rel.text.* .rel.gnu.linkonce.t.*)
        *(.rel.fini)
        *(.rel.rodata .rel.rodata.* .rel.gnu.linkonce.r.*)
        *(.rel.data .rel.data.* .rel.gnu.linkonce.d.*)
        *(.rel.tdata .rel.tdata.* .rel.gnu.linkonce.td.*)
        *(.rel.tbss .rel.tbss.* .rel.gnu.linkonce.tb.*)
        *(.rel.ctors)
        *(.rel.dtors)
        *(.rel.got)
        *(.rel.bss .rel.bss.* .rel.gnu.linkonce.b.*)
    }
    .rela.dyn    :
    {
        *(.rela.init)
        *(.rela.text .rela.text.* .rela.gnu.linkonce.t.*)
        *(.rela.fini)
        *(.rela.rodata .rela.rodata.* .rela.gnu.linkonce.r.*)
        *(.rela.data .rela.data.* .rela.gnu.linkonce.d.*)
        *(.rela.tdata .rela.tdata.* .rela.gnu.linkonce.td.*)
        *(.rela.tbss .rela.tbss.* .rela.gnu.linkonce.tb.*)
        *(.rela.ctors)
        *(.rela.dtors)
        *(.rela.got)
        *(.rela.bss .rela.bss.* .rela.gnu.linkonce.b.*)
    }
    .rel.plt     : { *(.rel.plt) }
```

```

.rela.plt      : { *(.rela.plt) }
.init         :
{
    KEEP (*(init))
} =0x90909090
.plt          : { *(.plt) }
.text         :
{
    *(.text .stub .text.* .gnu.linkonce.t.*)
    /* .gnu.warning bölümleri özellikle elf32.em. tarafından yönetilir */
    *(.gnu.warning)
} =0x90909090
.fini         :
{
    KEEP (*(fini))
} =0x90909090
PROVIDE (__etext = .);
PROVIDE (_etext = .);
PROVIDE (etext = .);
.rodata       : { *(.rodata .rodata.* .gnu.linkonce.r.*) }
.rodata1      : { *(.rodata1) }
.eh_frame_hdr : { *(.eh_frame_hdr) }
.eh_frame     : ONLY_IF_RO { KEEP (*(eh_frame)) }
.gcc_except_table : ONLY_IF_RO { *(.gcc_except_table) }
/* Veri bölütü için adresi ayarla. Bir sonraki sayfa yukarıda sayfadaki
   aynı adrese kadar ayarlamak istiyoruz. */
. = ALIGN (0x1000) - ((0x1000 - .) & (0x1000 - 1)); . = DATA_SEGMENT_ALIGN
(0x1000, 0x1000);
/* aşağıdaki *_array* bölümünü desteklemeyen araçların (tools)
   geriye dönük uyumluluğu için, glibc'mizin crt dosyaları başvurdukları
   sembollerin zayıf tanımlamalarını içerir. Kesinlikle zorunlu
   olmadıkça onları kullanmak istemiyoruz, çünkü bize boş bölümler
   getirirler; aşağıdaki PROVIDE'dan farklı olarak.
   Bu yüzden crt dosyalarından olan bölümleri çıkartırız. */
/DISCARD/ : {
    /*crti.o(.init_array .fini_array .preinit_array)
    /*crtn.o(.init_array .fini_array .preinit_array)
}
/* __preinit_array_start etiketinin uygun bir şekilde hizalandığından emin
   olun. Bunun yerine etiket tanımlamasını bölümün içine taşıyabilirdik fakat
   bağlayıcı ozaman boş olsa bile bölümü oluşturacaktı ki bu iyi değildir. */
. = ALIGN(32 / 8);
PROVIDE (__preinit_array_start = .);
.preinit_array : { *(.preinit_array) }
PROVIDE (__preinit_array_end = .);
PROVIDE (__init_array_start = .);
.init_array   : { *(.init_array) }
PROVIDE (__init_array_end = .);
PROVIDE (__fini_array_start = .);
.fini_array   : { *(.fini_array) }
PROVIDE (__fini_array_end = .);
.data         :
{
    *(.data .data.* .gnu.linkonce.d.*)
    SORT(CONSTRUCTORS)
}
.data1        : { *(.data1) }

```

```

.tdata      : { *(.tdata .tdata.* .gnu.linkonce.td.*) }
.tbss       : { *(.tbss .tbss.* .gnu.linkonce.tb.*) *(.tcommon) }
.eh_frame   : ONLY_IF_RW { KEEP (*(eh_frame)) }
.gcc_except_table : ONLY_IF_RW { *(.gcc_except_table) }
.dynamic    : { *(.dynamic) }
.ctors      :
{
    /* yapıcılarının başlangıcını bulmak için gcc crtbegin.o'yu
       kullanır, böylece biz bunun ilk olduğunu anlarız.
       Bu bir joker karakter (wildcard) olduğundan, kullanıcının
       crtbegin.o'ya karşı ilintilememesinin bir önemi yoktur.
       bağlayıcı bir joker karakteri eşleştirmek için bir dosya
       aramayacaktır. Joker karakter ayrıca crtbegin.o'nun hangi
       dizinde bulunduğu önemli olmadığı anlamına gelir. */
    KEEP (*crtbegin.o(.ctors))
    /* yapıcılarının (ctors) sıralanmasının bitişine kadar,
       crtend.o dosyasından .ctor bölümünü dahil etmek istemiyoruz.
       crtend dosyasındaki .ctor bölümü yapıcılarının sonu işaretini
       (end of ctors marker ) içerir ve bu sonuncu olmalıdır. */
    KEEP (*(EXCLUDE_FILE (*crtend.o ) .ctors))
    KEEP (*(SORT(.ctors.*)))
    KEEP (*(.ctors))
}
.dtors      :
{
    KEEP (*crtbegin.o(.dtors))
    KEEP (*(EXCLUDE_FILE (*crtend.o ) .dtors))
    KEEP (*(SORT(.dtors.*)))
    KEEP (*(.dtors))
}
.jcr        : { KEEP (*(.jcr)) }
.got        : { *(.got.plt) *(.got) }
_edata = .;
PROVIDE (edata = .);
__bss_start = .;
.bss        :
{
    *(.dynbss)
    *(.bss .bss.* .gnu.linkonce.b.*)
    *(COMMON)
    /* .bss bölümlerinin _end'e kadar olan boşlukları kullandığından emin olmak
       için buraya hizala. Herhangi bir girdi olmadığından dolayı .bss bölümleri
       görünmese bile hizalamanın doğru olduğundan emin olmak için .bss'den sonra
       hizala. */
    . = ALIGN(32 / 8);
}
. = ALIGN(32 / 8);
_end = .;
PROVIDE (end = .);
. = DATA_SEGMENT_END (.);
/* Stabs hata ayıklama bölümleri. */
.stab       0: { *(.stab) }
.stabstr     0: { *(.stabstr) }
.stab.excl   0: { *(.stab.excl) }
.stab.exclstr 0: { *(.stab.exclstr) }
.stab.index  0: { *(.stab.index) }
.stab.indexstr 0: { *(.stab.indexstr) }

```

```
.comment      0 : { *(.comment) }
/* DWARF hata ayıklama bölümleri.
   DWARF hata ayıklama bölümlerindeki semboller
   bölümün başına bağlandı, bu yüzden onları 0'dan başlatırız.*/
/* DWARF 1 */
.debug        0 : { *(.debug) }
.line         0 : { *(.line) }
/* GNU DWARF 1 eklentileri (extensions) */
.debug_srcinfo 0 : { *(.debug_srcinfo) }
.debug_sfnames 0 : { *(.debug_sfnames) }
/* DWARF 1.1 and DWARF 2 */
.debug_aranges 0 : { *(.debug_aranges) }
.debug_pubnames 0 : { *(.debug_pubnames) }
/* DWARF 2 */
.debug_info     0 : { *(.debug_info .gnu.linkonce.wi.*) }
.debug_abbrev   0 : { *(.debug_abbrev) }
.debug_line     0 : { *(.debug_line) }
.debug_frame    0 : { *(.debug_frame) }
.debug_str      0 : { *(.debug_str) }
.debug_loc      0 : { *(.debug_loc) }
.debug_macinfo  0 : { *(.debug_macinfo) }
/* SGI/MIPS DWARF 2 eklentileri */
.debug_weaknames 0 : { *(.debug_weaknames) }
.debug_funcnames 0 : { *(.debug_funcnames) }
.debug_typenames 0 : { *(.debug_typenames) }
.debug_varnames 0 : { *(.debug_varnames) }
}

=====
[root@localhost linux]#
```

## C. GRUB ve LILO

Hem **GNU GRUB**<sup>(B144)</sup> hem de **LILO**<sup>(B145)</sup> gerçek kipli çekirdek başlık biçimini tanır ve bootsect'i (tek sektör), kurulum kodunu (`setup_sects` sektörleri) ve sıkıştırılmış çekirdek imgesini (`syssize*16` bayt) belleğe yükler. Onlar yükleyici belirletecini (`type_of_loader`) doldurur ve çekirdeğe uygun parametreleri ve seçenekleri aktarmaya çalışır. İşlerini bitirdikten sonra denetim kurulum koduna geçer.

### C.1. GNU GRUB

Aşağıdaki GNU GRUB program taslağı grub-0.93 tabanlıdır.

```
stage2/stage2.c:main()
{
    -- run_menu()
    -- run_script();
    -- builtin = find_command(heap);
    -- kernel_func();           // "kernel" komutu için builtin->func()
    |   -- load_image();         // boot.c içinde BOOTSEC_SIGNATURE'ı ara
    |   /* 0x100000'dan başlayan bellek (bvmLinux, bbootsect, bsetup)
    |   * veya (vmlinux, bootsect, setup) sırasında onlar tarafından
    |   * doldurulur */
    -- initrd_func();           // "initrd" komutu için
    |   -- load_initrd();
    -- boot_func();             // örtük "boot" komutu için
}
```



```

        '-- linux_boot();          // stage2/asm.S içinde tanımlı
        or big_linux_boot();      // grub/asmstub.c! içinde değil

// stage2/asm.S içinde
linux_boot:
    /* çekirdeği kopyala */
    sistem kodunu 0x100000 adresinden 0x10000 adresine
    taşı (linux_text_len bytes);
big_linux_boot:
    /* gerçek kip kısmını kopyala */
    EBX = linux_data_real_addr;
    kurulum kodunu linux_data_tmp_addr (0x100000+text_len) adresinden
    linux_data_real_addr (0x9100 bytes) adresine taşı;
    /* %ebx değerini bölüt adresi olarak değiştir*/
    linux_setup_seg = (EBX >> 4) + 0x20;
    /* XXX çağıran işlevler için güvenli alandaki yeni yığıt göstericisi
    ESP = 0x4000;
    stop_floppy();
    /* linux önyüklemesi için son ayarlama */
    prot_to_real();
    cli;
    SS:ESP = BX:9000;
    DS = ES = FS = GS = BX;
    /* başlangıca sıçra, örn. ljmp linux_setup_seg:0
    * Dikkat ederseniz linux_setup_seg BX olarak değiştirildi. */
    .byte 0xea
    .word 0
linux_setup_seg:
    .word 0

```

GRUB kılavuzu için **info grub**'a bakın.

grub-0.93'ü uyarlıyorsanız ve **bsetup**'da değişiklik yapacaksanız **bildirilmiş tek GNU GRUB hatası**<sup>(B146)</sup> dikkate alınmalıdır.

## C.2. LILO

GRUB'dan farklı olarak, LILO sistemi önyüklerken yapılandırma dosyasını kontrol etmez. **lilo** uçbirimden çağırıldığında bazı trikler ortaya çıkar.

Aşağıdaki LILO program anahatları lilo-22.5.8 tabanlıdır.

```

lilo.c:main()
|-- cfg_open(config_file);
|-- cfg_parse(cf_options);
|-- bsect_open(boot_dev, map_file, install, delay, timeout);
|   |-- open_bsect(boot_dev);
|   '-- map_create(map_file);
|-- cfg_parse(cf_top)
|   '-- cfg_do_set();
|       '-- do_image();          // "image=" bölümü için walk->action
|           |-- cfg_parse(cf_image) -> cfg_do_set();
|           |-- bsect_common(&descr, 1);
|           |   |-- map_begin_section();
|           |   |-- map_add_sector(fallback_buf);
|           |   '-- map_add_sector(options);
|           |-- boot_image(name, &descr) veya boot_device(name, range, &descr);

```

```

|         |         |-- int fd = geo_open(&descr, name, O_RDONLY);
|         |         read(fd, &buff, SECTOR_SIZE);
|         |         map_add(&geo, 0, image_sectors);
|         |         map_end_section(&descr->start, setup_sects+2+1);
|         |         /* bsect_common() içinde iki sektör yaratıldı,
|         |         * bootsect için ise başka bir sektör*/
|         |         geo_close(&geo);
|         |         |-- fd = geo_open(&descr, initrd, O_RDONLY);
|         |         map_begin_section();
|         |         map_add(&geo, 0, initrd_sectors);
|         |         map_end_section(&descr->initrd, 0);
|         |         geo_close(&geo);
|         |-- bsect_done(name, &descr);
|-- bsect_update(backup_file, force_backup, 0); // önyükeme sektörünü güncelle
|-- make_backup();
|-- map_begin_section();
|   map_add_sector(table);
|   map_write(&param2, keytab, 0, 0);
|   map_close(&param2, here2);
|-- // ... önyükeme sektörünün yeniden yerleştirilmesini yap
|-- // ... bsect_wr'yi doğru yere ayarla
|-- write(fd, bsect_wr, SECTOR_SIZE);
|-- close(fd);

```

`map_register()` tüm kayıtlı sektörleri tanımlamakta kullanılan (CX, DX, AL) üçlüsü (veri yapısı `SECTOR_ADDR`) için bir liste tutarken `map_add()`, `map_add_sector()` ve `map_add_zero()` işlerini tamamlamak için `map_register()`'i çağırabilir.

LILO sistemi yüklemek için `first.S` ve `second.S`'yi çalıştırır. Eşlem (map) dosyasını, bootsect ve kurulum kodunu yüklemek için `second.S:doboot()` yöntemini çağırır. Sonra, sistem kodunu yüklemek için `lfile()` yöntemini çağırır, `launch2()` -> `launch()` -> `cl_wait()` -> `start_setup()` -> `start_setup2()` yöntemini çağırır ve son olarak ayar kodunu çalıştırmak için `jmp 0, SETUPSEG` komutunu çalıştırır.

LILO'nun ayrıntıları için "**man lilo**" ve "**man lilo.conf**"a bakınız.

### C.3. Kaynakça

- GNU GRUB<sup>(B147)</sup>
- GRUB Tutorial<sup>(B148)</sup>
- LILO (freshmeat.net)<sup>(B149)</sup>
- LDP HOWTO—INDEX: Boot Loaders and Booting the OS<sup>(B150)</sup>

## D. SSS

Henüz yok.

## GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation

to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ascii without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

### 3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### 4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

## 7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is

included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.



## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being  list their titles, with
the Front-Cover Texts being  list, and with the Back-Cover Texts
being  list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

## Notlar

Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.

(B3) <http://sf.linuxforum.net/projects/i386bc>

(B4) <http://www.geocities.com/feiyunw/linux/>

(B5) <http://members.comu.edu.tr/nyucel>

(B6) <http://lxr.linux.no/source?v=2.4.20>

(B7) <http://vim.sourceforge.net>

(B8) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_2.html#SEC3](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_2.html#SEC3)

(B9) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_3.html#SEC17](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_3.html#SEC17)

(B10) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_3.html#SEC10](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_3.html#SEC10)

(B11) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_3.html#SEC21](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_3.html#SEC21)

(B12) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_3.html#SEC14](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_3.html#SEC14)



- 
- (B13) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_3.html#SEC26](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_3.html#SEC26)
- 
- (B14) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_3.html#SEC18](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_3.html#SEC18)
- 
- (B15) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_3.html#SEC21](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_3.html#SEC21)
- 
- (1) **zImage** mahlası: sıkıştırılmış;
- 
- (2) **-C** makefile'ları okumadan önce dizin değiştirmek için kullanılan bir **make** komut satırı seçeneğidir.
- Bakınız **GNU make: Summary of Options** ([http://www.gnu.org/software/make/manual/html\\_chapter/make\\_9.html#SEC102](http://www.gnu.org/software/make/manual/html_chapter/make_9.html#SEC102)) ve **GNU make: Recursive Use of make** ([http://www.gnu.org/software/make/manual/html\\_chapter/make\\_5.html#SEC58](http://www.gnu.org/software/make/manual/html_chapter/make_5.html#SEC58)).
- 
- (B18) [http://www.gnu.org/software/binutils/manual/html\\_chapter/binutils\\_3.html](http://www.gnu.org/software/binutils/manual/html_chapter/binutils_3.html)
- 
- (3) "\$@" hedef anlamındadır, "\$<" ilk öngereksinim anlamındadır; Bakınız **GNU make: Automatic Variables** ([http://www.gnu.org/software/make/manual/html\\_chapter/make\\_10.html#SEC111](http://www.gnu.org/software/make/manual/html_chapter/make_10.html#SEC111));
- 
- (4) **--oformat binary** çalıştırılabilirin bellek dökümüne eşdeğer bir ham ikilik çıktı arar; Bakınız **Using LD, the GNU linker: Command Line Options** ([http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_2.html#SEC3](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_2.html#SEC3)).
- 
- (5) Hedef **vmlinux** burada **linux/Makefile** içinde tanımlanana göre değişiktir;
- 
- (6) "subst" bir MAKE işlevidir; Bakınız **GNU make: Functions for String Substitution and Analysis** ([http://www.gnu.org/software/make/manual/html\\_chapter/make\\_8.html#SEC85](http://www.gnu.org/software/make/manual/html_chapter/make_8.html#SEC85)).
- 
- (B23) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_3.html#SEC20](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_3.html#SEC20)
- 
- (B25) <http://tldp.org/HOWTO/Kernel-HOWTO/>
- 
- (B26) <http://www.gnu.org/software/make/manual/>
- 
- (B27) <http://www.gnu.org/software/binutils/manual/ld-2.9.1/>
- 
- (B28) <http://www.gnu.org/software/binutils/manual/>
- 
- (B29) <http://www.gnu.org/software/bash/manual/>
- 
- (B30) <http://www.ctyme.com/intr/rb-2445.htm>
- 
- (B31) <http://developer.intel.com/design/pentium4/manuals/>
- 
- (B32) <http://www.ctyme.com/intr/rb-0607.htm>
-

- (B34) <http://www.ctyme.com/intr/rb-0088.htm>
- (B35) <http://www.ctyme.com/intr/rb-0210.htm>
- (B36) <http://www.ctyme.com/intr/rb-0210.htm>
- (B37) <http://www.ctyme.com/intr/rb-0605.htm>
- (B43) <http://www.ctyme.com/intr/rb-0106.htm>
- (B44) <http://www.ctyme.com/intr/rb-0607.htm>
- (B45) <http://www.ctyme.com/intr/rb-0605.htm>
- (B46) <http://www.ctyme.com/intr/rb-1527.htm>
- (B48) <http://www.ctyme.com/intr/rb-0605.htm>
- (B49) <http://developer.intel.com/design/pentium4/manuals/>
- (B50) <http://www.cs.cmu.edu/~ralf/files.html>
- (B51) <http://www.gnu.org/software/grub>
- (B52) <http://freshmeat.net/projects/lilo>
- (B53) [http://ftp.us.xemacs.org/ftp/pub/linux/suse/suse/i386/7.3/dosutils/  
bootlin/technic.doc](http://ftp.us.xemacs.org/ftp/pub/linux/suse/suse/i386/7.3/dosutils/bootlin/technic.doc)
- (B54) <http://www.ctyme.com/intr/rb-0639.htm>
- (B55) <http://www.ctyme.com/intr/rb-0605.htm>
- (B56) <http://www.acpi.info>
- (B57) <http://www.ctyme.com/intr/rb-1741.htm>
- (B58) <http://www.ctyme.com/intr/rb-1739.htm>
- (B59) <http://www.ctyme.com/intr/rb-1529.htm>
- (B60) <http://www.ctyme.com/intr/rb-1757.htm>
- (B61) <http://www.ctyme.com/intr/rb-6135.htm>
- (B62) <http://www.ctyme.com/intr/rb-6184.htm>
- (B63) <http://www.ctyme.com/intr/rb-0639.htm>
- (B64) <http://www.ctyme.com/intr/rb-1594.htm>
- (B65) <http://www.ctyme.com/intr/rb-0575.htm>
- (B66) <http://www.ctyme.com/intr/rb-1394.htm>
- (B67) <http://www.ctyme.com/intr/rb-1398.htm>

---

(B68) <http://www.ctyme.com/intr/rb-1397.htm>

---

(B69) <http://www.ctyme.com/intr/rb-1397.htm>

---

(B70) <http://www.ctyme.com/intr/rb-1394.htm>

---

(B71) <http://www.ctyme.com/intr/rb-1398.htm>

---

(B72) <http://www.ctyme.com/intr/rb-1398.htm>

---

(B75) <http://www.win.tue.nl/~aeb/linux/kbd/A20.html>

---

(B76) <http://www.ctyme.com/intr/rb-1336.htm>

---

(B81) <http://www.ctyme.com/intr/rb-2273.htm>

---

(B82) <http://www.win.tue.nl/~aeb/linux/kbd/A20.html>

---

(B83) <http://www.student.cs.uwaterloo.ca/~cs452/postscript/book.ps>

---

(B84) <http://developer.intel.com/design/pentium4/manuals/>

---

(B85) [http://www.gnu.org/software/binutils/manual/gas-2.9.1/html\\_chapter/as\\_16.html#SEC205](http://www.gnu.org/software/binutils/manual/gas-2.9.1/html_chapter/as_16.html#SEC205)

---

(B86) <http://www.xml.com/ldd/chapter/book/ch13.html#t1>

---

(B87) [http://www.gnu.org/software/binutils/manual/gas-2.9.1/html\\_chapter/as\\_5.html#SEC48](http://www.gnu.org/software/binutils/manual/gas-2.9.1/html_chapter/as_5.html#SEC48)

---

(B89) [http://www.gnu.org/software/binutils/manual/ld-2.9.1/html\\_chapter/ld\\_2.html#SEC3](http://www.gnu.org/software/binutils/manual/ld-2.9.1/html_chapter/ld_2.html#SEC3)

---

(B91) <http://gcc.gnu.org/onlinedocs/gcc-3.3.2/gcc/Function-Attributes.html#Function%20Attributes>

---

(B92) <http://kernelnewbies.org/faq/index.php3#asmlinkage>

---

(B93) <http://www.ietf.org/rfc/rfc1952.txt>

---

(7) ID2 değeri gzip 0.5 için 158 (0x9e, \236) olabilir;

---

(8) XFL değeri 4 olduğunda ise sıkıştırıcı en hızlı algoritmayı kullanacaktır.

---

(9) FLG biti 0 olduğunda FTEXT, herhangi bir ek alan belirtmez.

---

(B95) <http://www.gzip.org>

---

(B96) <http://www.ietf.org/rfc/rfc1951.txt>

---

(B97) <http://www.gnu.org/software/binutils/manual/gas-2.9.1/>

---

(B98) <http://www.gnu.org/software/binutils/manual/ld-2.9.1/>

---

(B99) <http://developer.intel.com/design/pentium4/manuals/>

(B100) <http://www.gzip.org>

(B101) <http://freshmeat.net/projects/gzip>

(B102) <http://www.ietf.org/rfc/rfc1951.txt>

(B103) <http://www.ietf.org/rfc/rfc1952.txt>

(B104) <http://www.intel.com/design/pentium/datashts/242016.htm>

(B107) [http://www.gnu.org/software/binutils/manual/gas-2.9.1/html\\_chapter/as\\_7.html#SEC70](http://www.gnu.org/software/binutils/manual/gas-2.9.1/html_chapter/as_7.html#SEC70)

(B111) <http://developer.intel.com/design/pentium4/manuals/>

(B112) <http://www.intel.com/design/pentium/datashts/242016.htm>

(B113) <http://www.gnu.org/software/binutils/manual/gas-2.9.1/>

(B114) <http://www.gnu.org/software/binutils/manual/>

(B115) <http://kernelnewbies.org/faq/index.php3#asmlinkage>

(B116) <http://www.tldp.org/LDP/lki/lki-1.html#ss1.8>

(B117) <http://www.symonds.net/~abhi/files/mm/mm.html>

(B118) <http://linux-mm.org/docs/initialization.html>

(B119) <http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html>

(B120) <http://www.tldp.org/HOWTO/BogoMips.html>

(B122) <http://freshmeat.net/projects/sysvinit>

(B123) <http://www.tldp.org/LDP/lki/index.html>

(B124) <http://kernelnewbies.org/documents/>

(B125) <http://www.tldp.org/HOWTO/HOWTO-INDEX/index.html>

(B126) <http://www.xml.com/ldd/chapter/book>

(B127) <http://www.intel.com/design/pentium/datashts/242016.htm>

(B132) <http://www.intel.com/design/pentium/datashts/242016.htm>

(B133) <http://developer.intel.com/design/pentium4/manuals/>

(B134) <http://www.tldp.org/LDP/lki/lki-1.html#ss1.7>

(B135) <http://www.tldp.org/HOWTO/SMP-HOWTO.html>

(B136) <http://www.acpi.info>

- (B137) <http://www.redhat.com>
- (B138) <http://kernelnewbies.org/faq/index.php3#compile>
- (B139) <http://www.digitalhermit.com/linux/kernel.html>
- (B140) <http://www.debian.org>
- (B141) <http://www.debian.org/releases/stable/i386/ch-«post-«install.en.html#s-«kernel-«baking>
- (B142) <http://www.debian.org/doc/manuals/debian-«faq/ch-«kernel.en.html>
- (B143) <http://www.debian.org/doc/manuals/reference/ch-«kernel.en.html>
- (B144) <http://www.gnu.org/software/grub>
- (B145) <http://freshmeat.net/projects/lilo>
- (B146) <http://mail.gnu.org/archive/html/bug-«grub/2003-«03/msg00030.html>
- (B147) <http://www.gnu.org/software/grub/>
- (B148) <http://www.openbg.net/sto/os/xml/grub.html>
- (B149) <http://freshmeat.net/projects/lilo>
- (B150) <http://www.tldp.org/HOWTO/HOWTO-«INDEX/os.html#OSBOOT>

Bu dosya (linux-i386-boot-code-howto.pdf), belgenin XML biçiminin T<sub>E</sub>XLive ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

22 Ocak 2007