هسته‌ی لینوکس و کامپایل آن

علی موسوی

بهار ۱۳۹۳

فهرست مطالب

هسته چیست؟

هسته بخش اصلی اکثر سیستم‌عامل‌های مدرن است. برخی از سیستم‌های عامل ابتدایی فاقد هسته بودند که در این‌گونه سیستم‌ها کاربر مستقیماً با سخت‌افزار ارتباط برقرار می‌نمود. ارتباط مستقیم با سخت‌افزار گرچه باعث افزایش سرعت شده و همچنین دست کاربر را برای برقراری هرگونه ارتباطی با سخت‌افزارش باز می‌گذاشت، اما مشکلات بسیاری را به وجود می‌آورد. یک اشتباه از سوی کاربر و یا یکی از نرم‌افزارها، می‌توانست سخت‌افزار را دچار مشکل کند و همچنین مدیریت نرم‌افزارها برای استفاده از منابع سخت‌افزاری بسیار پیچیده بود.

برای رفع این‌گونه مشکلات، هسته‌ی سیستم‌عامل به وجود آمد و به عنوان یک رابط بین کاربر و سخت‌افزار طراحی و به یکی از اصلی‌ترین و اساسی‌ترین بخش‌های سیستم‌های عامل امروزی تبدیل شد.

وظایف هسته

۱- مدیریت قطعات:

به جز RAM و CPU قطعات بسیاری به یک کامپیوتر متصل می‌شوند. از جمله کارت‌های گرافیک، صدا، مودم، کارت شبکه و ...) که هر یک به نحوی کار می‌کنند و هسته با استفاده از درایورهای مختلف از نحوه‌ی کار آن‌ها مطلع شده و قادر به برقرار ارتباط مناسب بین قطعات می‌شود. هر ارتباطی که با سخت‌افزار صورت می‌گیرد باید از قواعد خاصی پیروی کند و هسته اطمینان پیدا می‌کند که این قواعد به درستی رعایت می‌شوند.

۲- مدیریت مموری (حافظه رم)

هر پروسه‌ای که اجرا می‌شود، نیاز به مقدار معینی از حافظه‌ی رم دارد. هسته اطمینان پیدا می‌کند که مقدار حافظه‌ی مورد نیاز هر پروسه، به آن اختصاص داده شود. هسته همچنین باید از تداخل اطلاعات ذخیره شده در حافظه جلوگیری نماید تا اطلاعات مربوط به یک پروسه، توسط پروسه‌ای دیگر دستکاری نشده و مشکلی برای پروسه‌ها به وجود نیاید.

۳- مدیریت CPU

هسته برای اطمینان از اینکه هر پروسه مدت زمان لازم را برای استفاده از CPU در اختیار داشته باشد، پروسه‌ها را اولویت‌بندی می‌نماید و زمان لازم را به هر پروسه اختصاص می‌دهد. این مدیریت تنها محدود به زمان نشده و عواملی مثل مجوزهای امنیتی هر پروسه، مالکیت (ownership) پروسه، ارتباطات بین پروسه‌های مختلف و ... را شامل می‌شود.

۴- ارتباط بین کاربر و سخت‌افزار

در نهایت کرنل وظیفه دارد بستری را برای دسترسی اطلاعات مختلف سخت‌افزاری، منابع سیستم و ... در اختیار برنامه‌نویسان و کاربران محیا نماید. برنامه‌نویسان می‌توانند با استفاده از درخواست‌های سیستمی (system calls) به این اطلاعات دسترسی پیدا کرده و در صورت نیاز تغییری در وضعیت سیستم خود به وجود آورند.

هسته لینوکس:

یکی از مهم‌ترین بخش‌های سیستم‌عامل لینوکس هسته‌ی آن است و هسته‌ی لینوکس، لینوکس است. بله... نام سیسنم‌عامل لینوکس از نام هسته‌ی آن گرفته شده است.

پروژه‌ی هسته‌ی لینوکس در سال ۱۹۹۱ و توسط «لینوس توروالدز» ایجاد شد و هنوز هم توسط او مدیریت می‌شود. کرنل لینوکس پس از انتشار اولین نسخه‌ی آن در سال ۱۹۹۴، به شدت گسترش پیدا کرد و گرچه عده‌ی انگشت‌شماری تصمیم می‌گیرند که چه کدهایی به هسته راه پیدا کند، اما اکنون بیش از صدها برنامه‌نویس برای هر نسخه‌ی هسته کد می‌نویسند.

به هسته‌ای که توسط تیم لینوس توروالدز منتشر می‌شود هسته‌ی وانیلی (vanilla kernel) می‌گویند. پس از انتشار هر نسخه از کرنل وانیلی، توزیع‌های لینوکس و پروژه‌های مختلف توسعه‌ی کرنل، تغییرات مورد نظر خود را در هسته‌ی وانیلی ایجاد کرده و برای کاربرانشان منتشر می‌کنند. این هسته‌ها معمولاً شامل امکاناتی می‌شوند که کرنل وانیلی هنوز نمی‌خواهد آن‌ها را پشتیبانی کند و یا به دلایلی پشتیبانی آن‌ها را متوقف کرده است.

چگونه هسته‌ی لینوکس را کامپایل کنیم؟

۱- جمع‌آوری اطلاعات سخت‌افزاری:

پیش از اینکه بتوانید کرنل خود را کامپایل کنید می‌بایست سخت‌افزارهای سیستم خود را بشناسید. به طور معمول کسی دقیقاً نمی‌داند که از چه سخت‌افزارهایی استفاده می‌کند. خوشبختانه لینوکس ابزارهای خوبی برای شناخت سخت‌افزارهای سیستم دارد که در زیر به معرفی برخی از مهم‌ترین آن‌ها می‌پردازیم:

آ. اطلاعات حافظه رم:

برای دانستن اطلاعات حافظه‌ی رم خود می‌توانید از دستور زیر استفاده کنید:

```
$ cat /proc/meminfo
MemTotal:        2030260 kB
MemFree:          230200 kB
MemAvailable:     604200 kB
Buffers:           93492 kB
Cached:           506440 kB
SwapCached:            8 kB
Active:          1146284 kB
Inactive:         533528 kB
Active(anon):     975512 kB
Inactive(anon):   220696 kB
Active(file):     170772 kB
Inactive(file):   312832 kB
Unevictable:       14236 kB
Mlocked:           14236 kB
HighTotal:       1149700 kB
HighFree:         101624 kB
LowTotal:         880560 kB
LowFree:          128576 kB
SwapTotal:       2157564 kB
SwapFree:        2157500 kB
…
```

همین‌طور که مشاهده می‌کنید، مقدار رم سیستم شما و سایر اطلاعات در رابطه با حافظه‌ی خود را می‌توانید در خروجی ببینید. برای به دست آوردن اطلاعت رم همچنین می‌توان از دستور free استفاده نمود.

ب. اطلاعات CPU

مشابه رم، برای به دست آوردن اطلاعات مربوط به CPU می‌توان از دستور زیر استفاده کرد:

```
$ cat /proc/cpuinfo
```

```
processor     : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 23
model name    : Intel(R) Core(TM)2 Duo CPU     T6500  @ 2.10GHz
stepping      : 10
microcode     : 0xa07
cpu MHz               : 1200.000
cache size    : 2048 KB
physical id   : 0
siblings      : 2
core id               : 0
cpu cores     : 2
apicid        : 0
initial apicid        : 0
fdiv_bug      : no
f00f_bug      : no
coma_bug      : no
fpu           : yes
fpu_exception : yes
cpuid level   : 13
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm constant_tsc
arch_perfmon pebs bts aperfmperf pni dtes64 monitor ds_cpl est tm2 ssse3 cx16
xtpr pdcm sse4_1 xsave lahf_lm dtherm
bogomips      : 4190.70
clflush size  : 64
cache_alignment       : 64
address sizes : 36 bits physical, 48 bits virtual
power management:
```

همان‌طور که می‌بینید، اطلاعات کاملی در رابطه با سی‌پی‌یو توسط این دستور به دست می‌آید.

پ. قطعات PCI

برای اطلاع از قطعات PCI متصل شده به دستگاهتان می‌توانید از دستور زیر استفاده کنید:

```
$ lspci -k
00:00.0 Host bridge: Intel Corporation Mobile 4 Series Chipset Memory Controller
Hub (rev 07)
       Subsystem: Dell Device 02aa
       Kernel driver in use: agpgart-intel
       Kernel modules: intel_agp
00:02.0 VGA compatible controller: Intel Corporation Mobile 4 Series Chipset
Integrated Graphics Controller (rev 07)
       Subsystem: Dell Device 02aa
       Kernel driver in use: i915
       Kernel modules: i915
00:02.1 Display controller: Intel Corporation Mobile 4 Series Chipset Integrated
Graphics Controller (rev 07)
       Subsystem: Dell Device 02aa
00:1a.0 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI
Controller #4 (rev 03)
       Subsystem: Dell Device 02aa
       Kernel driver in use: uhci_hcd
       Kernel modules: uhci_hcd
00:1a.1 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI
Controller #5 (rev 03)
```

```
	Subsystem: Dell Device 02aa
	Kernel driver in use: uhci_hcd
	Kernel modules: uhci_hcd
00:1a.2 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI
Controller #6 (rev 03)
	Subsystem: Dell Device 02aa
	Kernel driver in use: uhci_hcd
	Kernel modules: uhci_hcd
00:1a.7 USB controller: Intel Corporation 82801I (ICH9 Family) USB2 EHCI
Controller #2 (rev 03)
	Subsystem: Dell Device 02aa
	Kernel driver in use: ehci-pci
	Kernel modules: ehci_pci
00:1b.0 Audio device: Intel Corporation 82801I (ICH9 Family) HD Audio Controller
(rev 03)
	Subsystem: Dell Device 02aa
	Kernel driver in use: snd_hda_intel
	Kernel modules: snd_hda_intel
00:1c.0 PCI bridge: Intel Corporation 82801I (ICH9 Family) PCI Express Port 1
(rev 03)
	Kernel driver in use: pcieport
	Kernel modules: shpchp
00:1c.1 PCI bridge: Intel Corporation 82801I (ICH9 Family) PCI Express Port 2
(rev 03)
	Kernel driver in use: pcieport
	Kernel modules: shpchp
00:1c.2 PCI bridge: Intel Corporation 82801I (ICH9 Family) PCI Express Port 3
(rev 03)
	Kernel driver in use: pcieport
	Kernel modules: shpchp
00:1c.4 PCI bridge: Intel Corporation 82801I (ICH9 Family) PCI Express Port 5
(rev 03)
	Kernel driver in use: pcieport
	Kernel modules: shpchp
00:1d.0 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI
Controller #1 (rev 03)
	Subsystem: Dell Device 02aa
	Kernel driver in use: uhci_hcd
	Kernel modules: uhci_hcd
00:1d.1 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI
Controller #2 (rev 03)
	Subsystem: Dell Device 02aa
	Kernel driver in use: uhci_hcd
	Kernel modules: uhci_hcd
00:1d.2 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI
Controller #3 (rev 03)
	Subsystem: Dell Device 02aa
	Kernel driver in use: uhci_hcd
	Kernel modules: uhci_hcd
00:1d.7 USB controller: Intel Corporation 82801I (ICH9 Family) USB2 EHCI
Controller #1 (rev 03)
	Subsystem: Dell Device 02aa
	Kernel driver in use: ehci-pci
	Kernel modules: ehci_pci
00:1e.0 PCI bridge: Intel Corporation 82801 Mobile PCI Bridge (rev 93)
00:1f.0 ISA bridge: Intel Corporation ICH9M LPC Interface Controller (rev 03)
	Subsystem: Dell Device 02aa
	Kernel driver in use: lpc_ich
	Kernel modules: lpc_ich
```

```
00:1f.2 IDE interface: Intel Corporation 82801IBM/IEM (ICH9M/ICH9M-E) 2 port
SATA Controller [IDE mode] (rev 03)
       Subsystem: Dell Device 02aa
       Kernel driver in use: ata_piix
       Kernel modules: ata_piix, pata_acpi, ata_generic
00:1f.3 SMBus: Intel Corporation 82801I (ICH9 Family) SMBus Controller (rev 03)
       Subsystem: Dell Device 02aa
       Kernel driver in use: i801_smbus
       Kernel modules: i2c_i801
00:1f.5 IDE interface: Intel Corporation 82801IBM/IEM (ICH9M/ICH9M-E) 2 port
SATA Controller [IDE mode] (rev 03)
       Subsystem: Dell Device 02aa
       Kernel driver in use: ata_piix
       Kernel modules: ata_piix, pata_acpi, ata_generic
09:00.0 Ethernet controller: Marvell Technology Group Ltd. 88E8040 PCI-E Fast
Ethernet Controller (rev 13)
       Subsystem: Dell Device 02aa
       Kernel driver in use: sky2
       Kernel modules: sky2
0c:00.0 Network controller: Broadcom Corporation BCM4312 802.11b/g LP-PHY (rev
01)
       Subsystem: Dell Wireless 1397 WLAN Mini-Card
       Kernel driver in use: b43-pci-bridge
       Kernel modules: ssb
```

این دستور اطلاعات جامعی در رابطه با سخت‌افزار شما می‌دهد. مهم‌ترین این اطلاعات نام درایور و ماژولی که در هسته‌ی فعلی برای کار با هر سخت‌افزار استفاده شده است، می‌باشد. این اطلاعات یافتن درایور مورد نیاز هر سخت‌افزار را به هنگام تنظیم هسته برای شما ساده‌تر می‌کند.

ت. قطعات USB

دستور زیر اطلاعات مورد نیاز در مورد قطعات usb متصل به دستگاه را به شما می‌دهد:

```
$ lsusb
Bus 005 Device 003: ID 125f:a94a A-DATA Technology Co., Ltd.
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 002: ID 1bcf:0005 Sunplus Innovation Technology Inc.
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 004: ID 05ca:180a Ricoh Co., Ltd
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 004: ID 413c:8162 Dell Computer Corp. Integrated Touchpad
[Synaptics]
Bus 001 Device 003: ID 413c:8161 Dell Computer Corp. Integrated Keyboard
Bus 001 Device 002: ID 0a5c:4500 Broadcom Corp. BCM2046B1 USB 2.0 Hub (part of
BCM2046 Bluetooth)
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

۲- ماژول‌های هسته:

۱- ماژول چیست؟

هسته فایلی است که توسط راه‌انداز (boot loader) در مموری قرار می‌گیرد، اما لینوکس توانایی بارگزاری ماژول‌های هسته را نیز در حافظه، دارد.

ماژول هسته، بخشی از هسته است که می‌تواند به صورت پویا در حافظه قرار بگیرد و یا از حافظه خارج شود. یعنی شما می‌توانید هسته‌ای داشته باشید که از یک سخت‌افزار خاص پشتیبانی کند اما تا زمانی که واقعاً به آن سخت‌افزار نیازی نباشد آن را در حافظه قرار ندهد. این کار باعث کوچکتر شدن حجم هسته می‌شود.

ماژول‌ها معمولاً برای قطعات قابل شدن (مثل دستگاه‌های USB) استفاده می‌شوند، ولی توزیع‌های مختلف لینوکس که قصد پشتیبانی سخت‌افزارهای متفاوت کاربرانشان را دارند معمولاً درایور بسیاری از سخت‌افزارها را به صورت ماژول تنظیم می‌کنند و به هنگام بالاآمدن سیستم، ماژول‌های مورد نیاز را در حافظه بارگزاری می‌کنند. لینوکس به صورت خودکار سخت‌افزارهای سیستم را شناسایی نموده و ماژول درایور مورد نیاز هر کدام را بارگزاری می‌کند.

۲- Initial Ram Disk

مهم است بدانیم که ماژول‌ها خود در جایی ذخیره شده و از آنجا بارگزاری می‌شوند. بنابراین شما نمی‌توانید ماژول درایور یک دیسک سخت را در همان دیسک سخت قرار دهید و انتظار داشته باشید که لینوکس آن را بارگزاری نماید. و یا قراردادن ماژول درایور سیستم‌فایلی ext3 در یک پارتیشن ext3 کار نادرستی است. برای اینکه بتوان از چنین ماژول‌هایی می‌بایست از initial ram disk یا initrd استفاده کرد. Initrd فایلیست که چنین ماژول‌هایی را در خود جا می‌دهد و توسط راه‌انداز در حافظه قرار می‌گیرد تا هسته بتواند در صورت نیاز از این فایل استفاده کند.

لازم به ذکر است به جز مواردی که نیاز به پشتیبانی از طیف زیادی از سخت‌افزارها وجود داشته باشد، هسته‌ای که خوب کامپایل شود، نیازی به initrd ندارد. گرچه هسته‌ی لینوکس به صورت پیش‌فرض قادر به ایجاد فایل initrd می‌باشد.

۳- کار با ماژول‌ها

مهم‌ترین دستورات برای کار با ماژول‌ها lsmod، rmmod و modprobe هستند. برای لیست کردن ماژول‌هایی که در حال حاضر برای سیستم شما بارگزاری شده‌اند از دستور lsmod استفاده می‌شود:

```
$ lsmod
Module                  Size  Used by
fuse                   71217  3
joydev                  7691  0
arc4                    1596  2
b43                   347917  0
bcma                   31394  1 b43
mac80211              446215  1 b43
cfg80211              388522  2 b43,mac80211
rng_core                2888  1 b43
mousedev                9164  0
ums_realtek             6579  0
hid_generic              781  0
pcspkr                  1519  0
iTCO_wdt                4599  0
gpio_ich                3633  0
iTCO_vendor_support     1577  1 iTCO_wdt
dell_wmi                1249  0
dell_laptop            11305  0
sparse_keymap           2742  1 dell_wmi
led_class               2699  2 b43,dell_laptop
rfkill                 12995  3 cfg80211,dell_laptop
```

خروجی این دستور نام ماژول، فضایی که ماژول مصرف می‌کنند و تعداد/نام ماژول‌هایی که به آن

وابسته‌اند را نمایش می‌دهد.

برای این که یک ماژول را از حافظه خارج کنید، ابتدا مطمئن شوید که ماژول دیگری به آن وابسته نیست و سپس به صورت زیر عمل کنید:

```
# rmmod iTCO_wdt
```

و برای بارگزاری یک ماژول:

```
# modprobe iTCO_wdt
```

یکی از فواید استفاده از ماژول‌ها در هسته این است که می‌توان در صورت نیاز برخی پارامترهای یک ماژول را به هنگام بارگزاری آن تغییر داده و نحوه‌ی کار ماژول را با استفاده از پارامترهایی که هر ماژول پشتیبانی می‌کند، تحت تأثیر قرار داد. با دستور زیر می‌توان اطلاعاتی درباره‌ی هر ماژول به دست آورد:

```
$ modinfo uvcvideo
filename:       /lib/modules/3.14.3-1-
ARCH/kernel/drivers/media/usb/uvc/uvcvideo.ko.gz
version:        1.1.1
license:        GPL
description:    USB Video Class driver
author:         Laurent Pinchart <laurent.pinchart@ideasonboard.com>
srcversion:     B6EAA5A26874F36D613FD21
...
depends:        videodev,videobuf2-core,usbcore,media,videobuf2-vmalloc
intree:         Y
vermagic:       3.14.3-1-ARCH SMP preempt mod_unload modversions 686
parm:           clock:Video buffers timestamp clock
parm:           nodrop:Don't drop incomplete frames (uint)
parm:           quirks:Forced device quirks (uint)
parm:           trace:Trace level bitmask (uint)
parm:           timeout:Streaming control requests timeout (uint)
```

و در نهایت می‌توان ماژول را با پارامتر مورد نیاز بارگزاری کرد:

```
# modprobe uvcvideo nodrop=1
```

البته می‌توان برای همیشگی کردن یک پارامتر، آن را به همراه نام ماژول در یک فایل در فولدر / etc/modprobe.d/ قرار داد:

```
# cat /etc/modprobe.d/uvcvideo
options uvcvideo nodrop=1
```

۳- تنظیم هسته

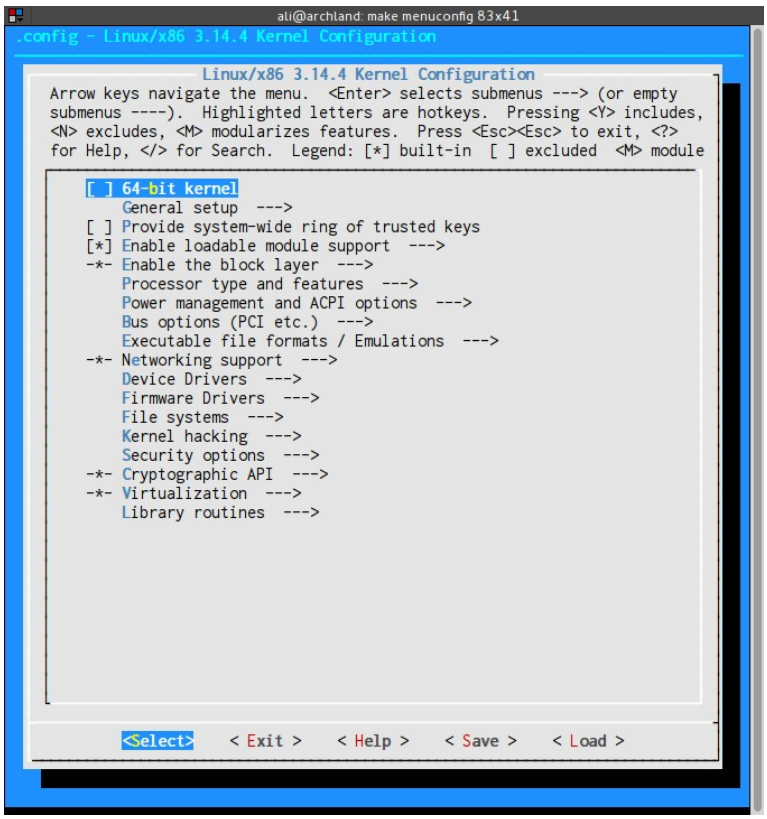
همیشه گفته می‌شود که کامپایل کردن هسته زمان بسیاری را می‌برد. اما بیشتر زمانی که برای کامپایل هسته سپری می‌شود، به هنگام تنظیم کردن آن است. خوشبختانه تنظیمات قابل ذخیره‌سازی و استفاده‌ی مجدد بوده و نیازی نیست که هر بار مجدداً زمان زیادی را برای تنظیم هسته سپری کنیم.

برای تنظیم هسته بد نیست نگاهی به وبسایت آقای «Pappy McFae» به آدرس http://www.kernel-seeds.org بیاندازید. ایشان یکی از کاربران توزیع جنتو هستند که منابع خوبی برای توضیح تنظیمات مختلف کرنل ایجاد کرده‌اند.

آ. ابزار تنظیم هسته

هسته به صورت پیش‌فرض دارای ابزاری برای تنظیم آن است. اگر به محل سورس هسته‌ی خود بر روی

هارد دیسکتان بروید و دستور make menuconfig را وارد کنید. ابزار تنظیم هسته اجرا خواهد شد و شما با گزینه‌های بسیار زیادی برای تنظیم بخش‌ها و قابلیت‌های مختلف هسته روبرو خواهید شد.



این ابزار کاری بیش از ویرایش یک فایل متنی به نام config. را که در محل سورس کرنل ایجاد می‌شود، بر عهده ندارد. با این حال ویرایش دستی این فایل به هیچ وجه توصیه نمی‌شود و شما می‌توانید اگر قبلاً یک فایل config. را ایجاد کرده‌اید پیش از تنظیم مجدد از آن بک‌آپ بگیرید تا مطمئن شوید مشکلی به وجود نخواهد آمد.

ب. چند نکته پیش از تنظیم کردن

.۱

از هسته‌ی لینوکس ورژن ۲.۶.۳۲ به بعد، آپشن‌های localmodconfig و localyesconfig به هسته اضافه شده است. همان‌طور که پیش‌تر گفته شد هسته به صورت پیش‌فرض سخت‌افزار را شناسایی و ماژول‌های مورد نیاز را لود می‌کند. با استفاده از آپشن localmodconfig می‌توان سخت‌افزارهای شناسایی شده را به عنوان ماژول در هسته تنظیم کرد تا فایل config. ما چیزی برای شروع تنظیمات داشته باشد.

```
# make localmodconfig
```

با استفاده از localyesconfig نیز می‌توان سخت‌افزارهای شناسایی شده را در خود هسته کامپایل نمود.

.۲

به هنگام کار با ابزار تنظیم هسته، می‌توان از کلید ? برای مشاهده‌ی راهنمای هر گزینه استفاده کرد. و همچنین برای جستجوی یک عبارت خاص می‌توان از کلید / استفاده نمود.

۴- کامپایل هسته

پس از اینکه کار تنظیم هسته شما تمام شد با استفاده از گزینه exit و تأیید اینکه قصد ذخیره‌سازی تنظیمات را دارید به محیط ترمینال خود برگشته و برای کامپایل هسته دستور زیر را وارد کنید.

```
$ make
```

کامپایل هسته با توجه به قدرت سی‌پی‌یو شما و نحوه‌ی تنظیم شما ممکن است از ۱۵ دقیقه تا بیش از دو ساعت به طول انجامد. پس از اتمام این مرحله کرنل کامپایل شده در با توجه به معماری سیستم شما (۶۴ بیتی یا ۳۲ بیتی) در هارد دیسک شما ذخیره می‌شود. به عنوان مثال در یک سیستم ۳۲ بیتی در آدرس arch/i386/boot/bzImage می‌توان هسته کامپایل شده را یافت.

به صورت پیش‌فرض دستور make مراحل کامپایل را یک به یک انجام می‌دهد. با این حال بیشتر سیستم‌ها دارای سی‌پی‌یوهای چند هسته‌ای می‌باشند که قابلیت کامپایل موازی را به کاربر می‌دهد. برای اینکه از این قابلیت استفاده کنید می‌توانید با استفاده از دستور پارامتر j- و تعداد هسته‌های سی‌پی‌یو خود اجرای موازی این دستور را ممکن ساخته و سرعت کامپایل هسته را افزایش دهید. به عنوان مثال برای یک سی‌پی‌یو ۴ هسته‌ای می‌توان دستور زیر را اجرا کرد:

```
$ make -j4
```

پس از کامپایل هسته، نوبت به نصب ماژول‌ها می‌شود. دستور زیر را اجرا کنید تا ماژول‌ها در محل مناسب نصب شوند. (/lib/modules/[kernelversion]/)

```
# make modules_install
```

در نهایت هسته‌ی کامپایل شده را به آدرس boot/ کپی کنید.

```
# cp arch/boot/i386/bzImage /boot/kernel-3
```

۵- تنظیم گراب

برای تنظیم گراب کافی است یک بار دستور آپدیت تنظیمات گراب را وارد نمایید تا گراب به صورت خودکار کرنل جدید را شناسایی و آن را در منوی راه‌انداز شما قرار دهد:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

۶- کامپایل مجدد هسته

تصور کنید که هسته شما با موفقیت کامپایل شده و با رضایت از آن استفاده می‌کنید اما پس از مدتی نیازمند به روز رسانی هسته خود می‌شوید. در این حالت نیازی به اجرای مجدد همه‌ی موارد بالا نیست و می‌توان با چند دستور ساده تنظیمات هسته فعلی خود را استفاده کنید. ابتدا به سورس هسته‌ی خود رفته و دستور زیر را اجرا کنید:

```
# zcat /proc/config.gz .config
```

فایل config.gz به صورت پیش‌فرض در هسته کامپایل شده و شامل تنظیمات هسته شما می‌باشد. با اجرای این دستور می‌توانید تنظیمات قبلی ا به شاخه‌ی هسته‌ی جدید خود کپی کنید و سپس با استفاده از دستور زیر از آن بهره‌برده کنید.

```
$ make oldconfig
```

اجرای این دستور باعث می‌شود که فقط در مورد تنظیماتی که تغییر کرده‌اند شما سؤال کند.

پس دستور بالا مراحل make و make modules_install را انجام داده و گراب خود را به روز رسانی کنید.