www.SoftGozar.com

# کامپایل کردن

# هسته‌ی لینوکس

## (Linux Kernel Compiling)

**مسعود باقری**

**گروه کامپیوتر دانشگاه بوعلی سینا**

www.SoftGozar.com

اَللهُ لا اِلهَ اِلّا هُوَ الحَیُّ القَیُّومُ
نَزَّلَ عَلَیکَ الکِتابَ بِالحَقِّ مُصَدِّقاً
لِما بَینَ یَدَیهِ
وَ اَنزَلَ التَّوراتَ وَ الاِنجیلَ

۲ و ۳ - آل‌عمران

www.SoftGozar.com

آن را که به صحرای علل تاخته‌اند
بی او همه کارها بپرداختـــــه‌اند
امـــروز بـهانه‌ای در انـداختـــه‌اند
فردا همه آن بود که درسـاخته‌اند

«حکیم خیّام»

# پیشگفتار

سیستم عامل[1] به عنوان کلیدی‌ترین نرم‌افزار سیستم‌های کامپیوتری، نقـش بـی‌بـدیلی در پیونـد دادن نرم‌افزارهای مختلف با سخت‌افزار سیستم، بر عهده دارد. اما، سیستم عامل خود دارای هسته‌ای است که از آن به کرنل[2] یاد می‌شود و در واقع وظیفه‌ی ارتباط بین نرم‌افزار و سخت افزار، و نیز شناسایی قطعات سخت‌افـزاری، و برخی وظایف دیگر، بر عهده‌ی این قسمت از سیستم عامل است.

لذا بر آن شدیم تا در این ارائه، علاوه بر معرفی اجمالی کرنل، نحوه‌ی به روز رسانی[3] هـسته‌ی سیـستم عامل لینوکس[4] را، که در حال حاضر از قوی‌ترین و حرفه‌ای‌ترین سیـستم‌هـای عامـل اسـت، بـه صـورت کـاملاً تفصیلی در اختیار خواننده‌ی گرامی قرار دهیم.

اسفند ۱۳۸۸

1. Operating System (OS)
2. Kernel
3. Update
4. Linux

www.SoftGozar.com

# فهرست مطالب

**عنوان** **صفحه**

**چکیده** **۵**
لغات کلیدی ۵

**فصل اول: کرنل** **۶**
۱-۱ : معرفی کرنل ۶
۱-۲ : ماجول‌های کرنل ۶
۱-۳ : چرا کامپایل؟ ۷
۱-۴ : نامگذاری کرنل لینوکس ۸

**فصل دوم: نحوه‌ی کامپایل کرنل** **۹**
۲-۱ : ملزومات اولیه ۹
۲-۲ : عملیات اصلی ۱۰
۲-۲-۱ : پیکربندی هسته ۱۰
۲-۲-۲ : ایجاد و نصب هسته ۱۱
۲-۲-۳ : کامپایل و نصب ماجول‌ها و ساخت کرنل ۱۱
۲-۲-۴ : RAM Disk و نحوه‌ی ایجاد آن ۱۱
۲-۲-۵ : به روز رسانی grub ۱۲

**منابع** **۱۳**

www.SoftGozar.com

# چکیده

کرنل یا هسته‌ی سیستم عامل، به عنوان بخشی از این نرم‌افزار، به هنگام راه‌اندازی[1] کامپیوتر، به همراه آن بارگذاری[2] شده و تا زمانی که سیستم روشن است در حافظه‌ی اصلی[3] مقیم می‌ماند. در حقیقت، این کرنل است که سخت‌افزارهای سیستمی[4] را شناسایی نموده و امکان استفاده از آنها را برای کاربر فراهم می‌کند. اما وظایف کرنل به این مورد محدود نمی‌شود و علاوه بر شناسایی ادوات سخت‌افزاری اصلی و جانبی، مدیریت مکانیزم‌های امنیتی و اولویت‌بندی اجرای برنامه‌ها را نیز بر عهده دارد.

به طور کلی، کرنل در پایین‌ترین سطح نرم‌افزاری و در ارتباط مستقیم با سخت‌افزار قرار دارد، به طوریکه کلیه‌ی نرم‌افزارهای کاربردی[5] و سیستمی[6]، از طریق کرنل با سخت‌افزار ارتباط برقرار می‌نمایند؛ این در حالی است که کاربر به ندرت مستقیماً با کرنل در ارتباط است. به عبارتی کرنل نقش هدایتگر سیستم را دارد و این امر اهمیت آن را به وضوح هویدا می‌سازد.

در این مکتوب، پس از معرفی مختصر کرنل، مسئله‌ی به روز رسانی هسته‌ی سیستم عامل لینوکس[7] (اوبونتو[8] ۹,۰۴)، مطرح می‌گردد و سعی بر آن است که با توضیحات کامل و گویا، نحوه‌ی انجام این کار، نه فقط به صورت تئوری، بلکه به صورت یک دستور کار عملی، ارائه گردد.

**لغات کلیدی:**

| | |
|---|---|
| Operating System | سیستم عامل |
| UBUNTU | نسخه‌ی اوبونتو از سیستم عامل لینوکس |
| Linux Kernel Updating | به روز رسانی هسته‌ی لینوکس |

1. start up
2. load
3. RAM
4. system hardwares
5. application softwares
6. system softwares
7. kernel updating of Linux Operating System
8. UBUNTU

www.SoftGozar.com

# ۱ کرنل

## ۱-۱ : معرفی کرنل

کرنل به هسته‌ی هر سیستم عامل اتلاق می‌شود. یک کرنل، یک جزء پیچیده‌ی نرم‌افزاری است که فرآیندها و تعاملات میان آنها را که در داخل سیستم عامل رخ می‌دهند، مدیریت می‌کند. به عبارت ساده‌تر، کرنل پایین‌ترین سطح نرم‌افزار در سیستم شماست و نقش هدایت‌کننده را برای همه‌ی سیستم بازی می‌کند. دسترسی به درایورهای سخت‌افزاری، مکانیزم‌های امنیتی، شبکه و خیلی چیزهای دیگر را مدیریت می‌کند؛ در مورد اولویت اجرای برنامه‌ها تصمیم می‌گیرد و تقریباً برای همه‌ی نرم‌افزارها، با سخت‌افزار در سیستم ارتباط برقرار می‌کند. شما به عنوان یک کاربر، به ندرت مستقیماً با کرنل ارتباط برقرار می‌کنید و در عوض با برنامه‌های کاربردی که کرنل آنها را مدیریت می‌کند، کار می‌کنید.

## ۱-۲ : ماجول[1]های کرنل

کرنل لینوکس از قسمت‌هایی تشکیل شده است که اصطلاحاً به آنها module یا اگر بخواهیم کامل‌تر بگوییم kernel module گفته می‌شود. بسیاری از این ماجول‌ها، درایورهای مختلف هستند. ماجول‌ها به گونه‌ای طراحی شده و در کنار بدنه‌ی اصلی کرنل قرار گرفته‌اند که به راحتی می‌توان یک ماجول را به کرنل اضافه نمود یا ماجولی را از کرنل کنار گذاشت.

بالاتر اشاره شد که کرنل به عنوان هسته‌ی سیستم عامل تلقی می‌شود و لذا بالا آمدن سیستم عامل در یک کامپیوتر، منوط به بار[2] شدن کرنل آن است. از طرفی دانستیم که کرنل وظیفه‌ی هدایت همه‌ی سیستم، از جمله درایورهای سخت‌افزاری، مکانیزم‌های امنیتی و شبکه را بر عهده دارد؛ به طوریکه برای این‌که سیستم بتواند با هر کدام از این ادوات ارتباط داشته‌باشد، باید ماجول مربوط به آن، به کرنل اضافه شده باشد.

وقتی ماجولی به کرنل اضافه می‌شود، در حقیقت به عنوان جزئی از آن محسوب می‌گردد و لذا به هنگام بالا آمدن[3] سیستم عامل، ماجول‌های اضافه‌شده به کرنل به عنوان قسمتی از کرنل به همراه آن بارگذاری می‌شوند. از سوی دیگر می‌دانیم که سیستم عامل به عنوان نرم‌افزار اصلی سیستم، در تمام مدت روشن بودن سیستم در حافظه‌ی اصلی مقیم می‌شود و لذا همواره قسمتی از حافظه در اشغال سیستم عامل است. چون کرنل قسمتی از سیستم عامل است، پس بخشی از این فضای اشغال شده مربوط به کرنل است. در عین حال احتمالاً ماجول‌هایی به کرنل اضافه شده‌است که باعث بیشتر شدن اندازه‌ی آن و در نتیجه سیستم عامل می‌شود که نهایتاً منجر به اشغال شدن فضای بیشتری از حافظه توسط سیستم عامل می‌گردد.

---

1. module
2. load
3. start up

www.SoftGozar.com

در کرنل‌هایی که برای استفاده‌ی عموم آماده شده‌اند، اکثر ماجول‌ها، که یا برای شناساندن قطعات سخت‌افزاری مختلف، یا برای آماده کردن سیستم عامل جهت ارتباط برقرار کردن با نرم‌افزارهای خاصی (مثلاً نرم‌افزارهای گرافیکی) طراحی شده‌اند، به کرنل اضافه شده‌اند؛ به‌طوریکه کرنل مورد نظر انواع درایورهای سخت-افزاری موجود (حتی درایورهای مشابه از کمپانی‌های مختلف) و مکانیسم‌های مختلف (که ممکن است هرگز روی آن سیستم مورد نیاز نباشند) را پشتیبانی می‌کند.

این ویژگی از یک لحاظ مفید است، چرا که به این ترتیب سیستم عامل اغلب ادوات و درایورهای سیستم را بدون نیاز به نصب نرم‌افزارهای مربوط، شناسایی خواهد کرد و به عبارتی کار کاربر را از این نظر، تا حد زیادی ساده می‌کند. اما از لحاظ دیگر ویژگی مناسبی محسوب نمی‌شود؛ با اندکی تأمل درمی‌یابیم که عملاً درصد بسیار کمی از ماجول‌های اضافه شده به کرنل، روی سیستم مورد نظر کاربرد دارند و باقی ماجول‌ها، که بخش عمده‌ای از کرنل‌هایی که برای استفاده‌ی عموم تهیه شده‌اند را، تشکیل می‌دهند، در حقیقت در هدایت فرآیندهای عمومی سیستم بی‌تأثیرند و علی‌رغم اینکه حجم زیادی از فضای اشغال‌شده‌ی RAM توسط سیستم عامل، به آنها برمی‌گردد، تنها برای استفاده‌ی نرم‌افزارهای خاصی در این کرنل‌ها تعبیه شده‌اند و همانطور که قبلاً هم اشاره شد، خیلی از این ماجول‌ها اصلاً برای کاربر عادی کاربردی ندارند و حتی یک کاربر حرفه‌ای هم به همه‌ی آن‌ها نیاز ندارد.

## ۱-۳ : چرا کامپایل؟

برای بهینه کردن شرایط، جهت کوچک نمودن حجم کرنل و حذف ماجول‌های بلااستفاده، در سیستم عامل لینوکس این امکان فراهم شده‌است که کاربر بتواند کرنل را مطابق میل خود و بر اساس نیازهای واقعی سیستم خود تنظیم نماید. به عبارتی کاربر می‌تواند تا آن‌جا که بخواهد از شاخ و برگ‌های اضافه‌ی کرنل حذف نماید و هسته‌ی سیستم عامل را خودش پیکربندی[1] نماید. حالا به راحتی می‌توانیم پاسخ این سؤال را بدهیم که "چرا می‌خواهیم هسته را مجدداً کامپایل نماییم؟"

برای این‌که سیستم عامل Gnu/Linux بر روی انواع بسته‌های سخت‌افزاری قابل استفاده باشد، در هر توزیع[2] از این سیستم عامل، یک هسته‌ی عمومی[3] برای کامپیوتر شما در نظر گرفته می‌شود که طبعاً بهینه‌ترین حالت نمی‌باشد. بی‌شک مهم‌ترین دلیل کامپایل مجدد، بهینه شدن هسته در قبال نیازهای شما خواهد بود و شما در واقع سیستم عامل سفارشی برای خود می‌سازید. البته شما هر وقت که بخواهید می‌توانید ماجول‌ها را به کرنل اضافه نمایید یا آن‌ها را از کرنل حذف کنید، و این هم بستگی به شما دارد که به هنگام کامپایل کردن خود کرنل[4]، چه گزینه‌هایی را به عنوان ماجول در نظر بگیرید.

ویژگی‌هایی که به هنگام کامپایل کرنل، مستقیماً در بدنه‌ی کرنل قرار داده می‌شوند، تا زمان کامپایل بعدی کرنل، همواره به عنوان جزء ثابتی از کرنل تلقی می‌شوند و دیگر نمی‌توانید آن‌ها را حذف کنید یا تغییری

1. configure
2. distribution
3. generalized Kernel

۴. کامپایل خود کرنل، یکی از مراحل کامپایل است که در ادامه توضیح داده می‌شود.

www.SoftGozar.com

در آن‌ها ایجاد نمایید، مگر این‌که مجدداً کرنل را کامپایل کنید؛ اما ویژگی‌هایی که به هنگام کامپایل کرنل به عنوان ماجول در نظر گرفته می‌شوند، جزء ثابتی از کرنل نبوده و بنا به صلاحدید ما می‌توانند به کرنل اضافه یا از آن کم شوند. خواهیم دید که ماجول‌ها درست بعد از کامپایل بدنه‌ی کرنل، کامپایل و نصب می‌شوند.

## ۱-۴ : نامگذاری کرنل لینوکس

نسخه[1]های مختلف کرنل لینوکس، در قالب یک سری عدد، مثلاً 2.6.15-23، مشخص می‌شوند. شماره-ی کرنل می‌تواند به ۴ قسمت شکسته شود:

- **شماره‌ی نسخه‌ی اصلی**[2]: این قسمت شماره‌ی نسخه‌ی اصلی کرنل است که در مورد این مثال برابر ۲ می‌باشد؛ به عبارتی، این بخش شامل شماره‌ی نگارش کرنل است و در حال حاضر کرنل‌ها با نگارش۲ موجودند.

- **شماره‌ی نسخه‌ی جزئی**[3]: این قسمت شماره‌ی نسخه‌ی جزئی کرنل است که در مورد این مثال برابر ۶ می‌باشد؛ به عبارتی، این شماره نشان‌دهنده‌ی نوع کرنل از جهت stable یا developmental بودن آن است.
  کرنل stable، کرنلی است که بدون خطا و ازکارافتادگی[4] ناگهانی یا اشکال مهم، تست شده است؛ درحالیکه کرنل‌های developmental پایداری کمتری دارند واحتمال ازکارافتادگی یا خطای بزرگ در آن‌ها بیشتر است. این دسته معمولاً با تغییراتی در کرنل‌های stable بدست می‌آیند.

- **شماره‌ی زیرسطح**[5]: این شماره می‌گوید کرنلی که با آن سروکار داریم کدام ویرایش[6] از بین کرنل‌هایی با شماره‌های نسخه‌ی اصلی و جزئی مشابه است. مثلاً در اینجا، تکرار[7] پنجم از کرنل‌های 2.6 را مثال زده‌ایم.

- **شماره‌ی سطح نسخه‌ی فوق‌العاده**[8]: این عدد مجموعه‌ای از patch ها و اضافات را مشخص می‌نماید که توسط مهندسان اوبونتو به کرنل اضافه می‌شود، تا کرنل مخصوص آن‌ها (و شما) کار کند را. هر مجموعه شماره‌گذاری می‌گردد و این شماره در نام کرنل نمایش داده می‌شود. در مثال فوق شماره‌ی مجموعه، 23 است.

---

1. version
2. major version
3. minor version
4. crash
5. sublevel number
6. release
7. iteration
8. extraversion level

www.SoftGozar.com

# ۲ نحوه‌ی کامپایل کرنل

در این قسمت، به تشریح چگونگی کامپایل کرنل سیستم عامل لینوکس (اوبونتو) می‌پردازیم. در ابتدا تمام آنچه را که در حین انجام عملیات کامپایل و به‌روزرسانی مورد نیاز است، تأمین نموده و سپس به سراغ مراحل اصلی کار می‌رویم.

## ۲-۱ : ملزومات اولیه

قبل از هر چیز باید شماره‌ی نسخه‌ی فعلی را بدانیم. برای رسیدن به این مطلوب از دستور:

```
#uname –r
```

استفاده می‌کنیم. حالا با دانستن شماره‌ی نسخه‌ی کرنل لینوکس خود، می‌توانید به آدرس http://www.kernel.org مراجعه نموده و نسخه‌ی مورد نظرتان را که به صورت فایل‌های tar.gz یا tar.bz2 است، دانلود کنید. فایل با پسوند gz. مربوط به بسته[1]ی نرم‌افزاری gzip است که تقریباً روی همه‌ی نسخه‌های لینوکس موجود است، و فایل با پسوند bz2. مربوط به بسته‌ی نرم‌افزاری جدیدتر bzip2 است که فشرده‌سازی[2] آن نسبت به gzip بهتر است.

قبل از شروع عملیات کامپایل، بهتر است که از همین‌جا با دستور:

```
#sudo bash
```

به عنوان کاربر root وارد عمل شویم. در ادامه، ابتدا باید فایل دانلود شده‌ی کرنل را به محل /usr/src/ منتقل و در آن‌جا، آن را از حالت بسته‌بندی[3] خارج نماییم[4]. فرض کنید اسم فایل دانلود شده، linux-2.6.31.7.tar.gz است. عملیات استخراج[5] را با دستور:

```
#tar –xzvf linux-2.6.31.7.tar.gz
```

انجام می‌دهیم. این دستور به صورت خودکار، یک پوشه[6] با نام linux-2.6.31.7 را، که همه‌ی آنچه را که مورد نیاز است در بردارد، در محل مذکور می‌سازد.

حالا همچنان که در مسیر /usr/src/ قرار داریم، باید یک پیوند نمادین[7] را ایجاد نماییم. این کار با حذف کردن محتویات فعلی پوشه‌ی src و پیوند دادن یک اشاره‌گر با نام کرنل جدید، به کرنل فعلی و به صورت زیر انجام می‌شود (نام فایل‌هایی که به عنوان آرگومان آمده‌اند، فرضی است):

```
#rm –f –r linux-headers-2.6.28-11
#rm –f –r linux-headers-2.6.28-11-generic
#ln –s linux-2.6.31.7 /usr/src/linux-headers-2.6.28-11
```

---

1. package
2. compressing
3. zipped
4. extract
5. extracting
6. folder
7. symbolic link

www.SoftGozar.com

دستور آخر، ln، برای ایجاد پیوند بین فایل‌ها به کار می‌رود و پارامتر s– مشخص می‌کند که این یک پیوند نمادین است.

## ۲-۲ : عملیات اصلی

در این مرحله که تا حدی هم وقت‌گیر و خسته‌کننده است، بدنه‌ی اصلی عملیات انجام می‌شود و طی آن کاربر به کرنل اختصاصی مورد نیاز خود دست می‌یابد؛ اما محقق شدن این امر منوط به اجرای صحیح عملیات است. شرح تفصیلی مراحل کار در ادامه آمده‌است.

### ۲-۲-۱ : پیکربندی هسته[1]

در این مرحله، هنوز در مسیر /usr/src/ قرار داریم، ولی حالا ابتدا با دستور:

```
#cd linux-headers-2.6.28.11
```

به محلی که اشاره‌گر پیوند نمادین مشخص می‌کند، می‌رویم و سپس با دستور:

```
#make config
```

وارد فاز پیکربندی کرنل می‌شویم. با اجرای این دستور، برنامه‌ی مربوط به پیکربندی کرنل، در محیط ترمینال[2] لینوکس بارگذاری[3] و آغاز می‌شود. توسط این برنامه تعداد نسبتاً زیادی سؤال از کاربر پرسیده و از او خواسته می‌شود که:

- درایورهای سخت‌افزار سیستم خود را از بین درایورهای مربوط به کمپانی‌های مختلف، مشخص کند.
- مکانیزم‌های مورد نیاز خود را از بین کلیه‌ی مکانیزم‌های موجود مشخص نماید.
- مشخص کند که کدام‌یک از موارد فوق به صورت ماجول به کرنل اضافه شود و کدام مستقیماً به بدنه‌ی کرنل ملحق گردد.

سؤالات تک‌به‌تک پرسیده می‌شوند و جواب اکثر آن‌ها به صورت گزینه‌هایی، جلوی آن‌ها قرار دارد و کاربر می‌تواند با وارد نمودن کاراکتر اول مربوط به گزینه‌ی مورد نظر خود، و فشردن کلید Enter انتخاب خود را انجام دهد. در مورد هر سؤال، یکی از گزینه‌ها (عمومی‌ترین آن‌ها) به عنوان انتخاب پیش‌فرض[4] در نظر گرفته شده و کاراکتر اول مربوط به آن با حرف بزرگ نوشته شده است. حال اگر در یک سؤال، بدون وارد نمودن کاراکتر، Enter را بزنیم، آن گزینه انتخاب می‌گردد. لازم به ذکر است که در مورد هیچ‌یک از سؤالات پرسیده‌شده، امکان بازگشت به عقب وجود ندارد، یعنی اگر اشتباهی رخ دهد باید تمامی مراحل از ابتدا تکرار شود.

با تمام شدن سؤالات، در واقع تنها سفارش ما حاضر شده و هنوز هیچ اتفاق دیگری نیفتاده است. حالا باید این سفارش را به سیستم تحویل دهیم تا بر اساس آن، کرنل دلخواه ما را آماده کند. این کار را در مراحل بعدی انجام می‌دهیم.

---

1. kernel configuration
2. terminal
3. load
4.default

www.SoftGozar.com

### ۲-۲-۲ : ایجاد و نصب هسته

اکنون زمان استراحت است و مراحل مشکل کار را به پایان رسانده‌ایم. قبل از اینکه بتوانیم عمل کامپایل را انجام دهیم، نیاز به تولید اطلاعات وابستگی[1] داریم و همچنین می‌بایست هرگونه فایل باقی مانده از کامپایل قبلی را نیز، حذف نماییم. با فرض اینکه کماکان در مسیر /usr/src/linux-headers-2.6.28-11 هستیم، نیازمندی‌های فوق را به‌ترتیب با استفاده از دستورات:

```
#make dep
```

و

```
#make clean
```

برآورده می‌سازیم. البته در هسته‌های جدید نیازی به make dep نداریم.

حالا نوبت به کامپایل هسته می‌رسد. از دستور:

```
#make bzImage
```

برای این منظور استفاده می‌کنیم. کامپایل توسط این دستور محقق می‌شود و پس از چند دقیقه، عملیات کامپایل هسته کامل شده و فایل bzImage در پوشه‌ی /usr/src/linux-headers-2.6.28-11/arch/x86/boot مشاهده می‌شود.

### ۲-۲-۳ : کامپایل و نصب ماجول‌ها و ساخت کرنل

اکنون زمان کامپایل نمودن ماجول‌هایی است که در مرحله‌ی پیکربندی آن‌ها را در نظر گرفتیم. این عمل با دستور:

```
#make modules
```

وقتی که در مسیر /usr/src/linux-headers-2.6.28-11 قرار داریم، انجام می‌گیرد. سپس با دستور:

```
#make modules_install
```

ماجول‌های کامپایل شده، در دایرکتوری /lib/modules/2.6.31.7/kernel/drivers نصب می‌شوند.

حالا همه چیز برای ساختن کرنل آماده است. دستور:

```
#make install
```

این کار را انجام می‌دهد. با این دستور، فایل bzImage ساخته شده، با نام vmlinuz-2.6.31.7، در مسیر /boot، کپی می‌شود.

### ۲-۲-۴ : RAM Disk و نحوه‌ی ایجاد آن

وقتی شما کامپیوتر را روشن می‌کنید، بایوس[2] قصد دارد کار را به هارد[3] بکشاند و کرنل را از روی هارد بخواند، اما چطور باید این کار را انجام دهد؟ مشکل این‌جاست، که این کرنل (یا احتمالاً ماجولی از کرنل) است که هارد را به سیستم می‌شناساند، و از طرفی می‌بایست کرنل به هنگام روشن شدن سیستم، از روی هارد به درون حافظه‌ی اصلی کپی شود، یعنی کرنل که باید هارد را به سیستم بشناساند، در ابتدا خودش تنها روی هارد قرار دارد. به عبارتی، علی‌الظاهر زمانی که سیستم را روشن می‌کنید، کرنلی در کار نیست، چرا که کرنل روی هارد است و هنوز هارد در دسترس سیستم عامل نیست. احتمالاً به عمق قضیه پی برده‌اید. شبیه مثال معروف

---

1. dependency information
2. BIOS
3. hard

www.SoftGozar.com

مرغ و تخم‌مرغ است، نه؟

برای حل این مشکل، initial ramdisk را ایجاد کرده‌اند. یعنی ابتدا در حافظه‌ی اصلی فایلی را ایجاد می‌نمایند و از آن به صورت یک فایل سیستم استفاده می‌کنند؛ یعنی کاری می‌کنند که یک تکه از RAM، به عنوان جزئی از هارد در نظر گرفته‌شود و از آن برای بارگذاری اولیه‌ی کرنل استفاده می‌شود. برای این کار دستور زیر را در حالیکه در مسیر boot/ هستیم، اجرا می‌کنیم:

mkinitramfs –o initrd.img-2.6.31.7 2.6.31.7

منظور از initrd در دستور فوق، همان فایل initial ramdisk است که بالاتر به آن اشاره شد. این فایل در واقع بارگذار اولیه[1]ی سیستم است.

## ۲-۲-۵ : به روز رسانی[2] grub

grub پوشه‌ای است که در آن اطلاعات مربوط به boot image های مختلف لینوکس می‌آید و لذا هر بار که کرنل جدیدی را کامپایل و نصب می‌کنیم، باید اطلاعات مربوط به آن را در این پوشه نیز قرار دهیم یا به عبارتی grub را به روز رسانی کنیم. این مرحله، آخرین گام ایجاد کرنل جدید است و برای آن لازم است که در مسیر boot/ قرار گرفته و دستور:

update-grub

را وارد کنیم.

حالا دیگر همه چیز آماده شده و کار ما نیز به اتمام رسیده است. در این مرحله می‌توانیم با رفتن به مسیر boot/grub/ و با استفاده از دستور:

nanu menu.lst

تغییرات صورت گرفته در grub را مشاهده کنیم.

اگر همه‌ی مراحل به درستی انجام شده‌باشند، اکنون می‌توانیم از لینوکس خارج شده و سیستم را با کرنل جدید راه‌اندازی[3] کنیم.

1. boot loader
2. update
3. start up

www.SoftGozar.com

# منابع

[1] Hudson Andry, Hudson Paul, "Ubuntu Unleashed 2008 Edition", Sams Publishing, United States of America, July 2008, Chapter 32, "Kernel and Modules Management".

[2] http://sinairv.blogspot.com/2007/05/blog-post_24.html, available on March 2010

[3] http://linuxboy.blogfa.com/post-16.aspx, available on March 2010

[4] http://forum.codecorona.com/forumdisplay.php?fid=13, available on March 2010

[5] http://asrejavan.org/showthread.php?t=348407, available on December 2009



www.SoftGozar.com