

2015

فرایند Boot در لینوکس (ویرایش سوم)

نویسنده: حسام الدین توحید



powered by

GNU/Linux

skywan13@chmail.ir





مقدمه مولف :

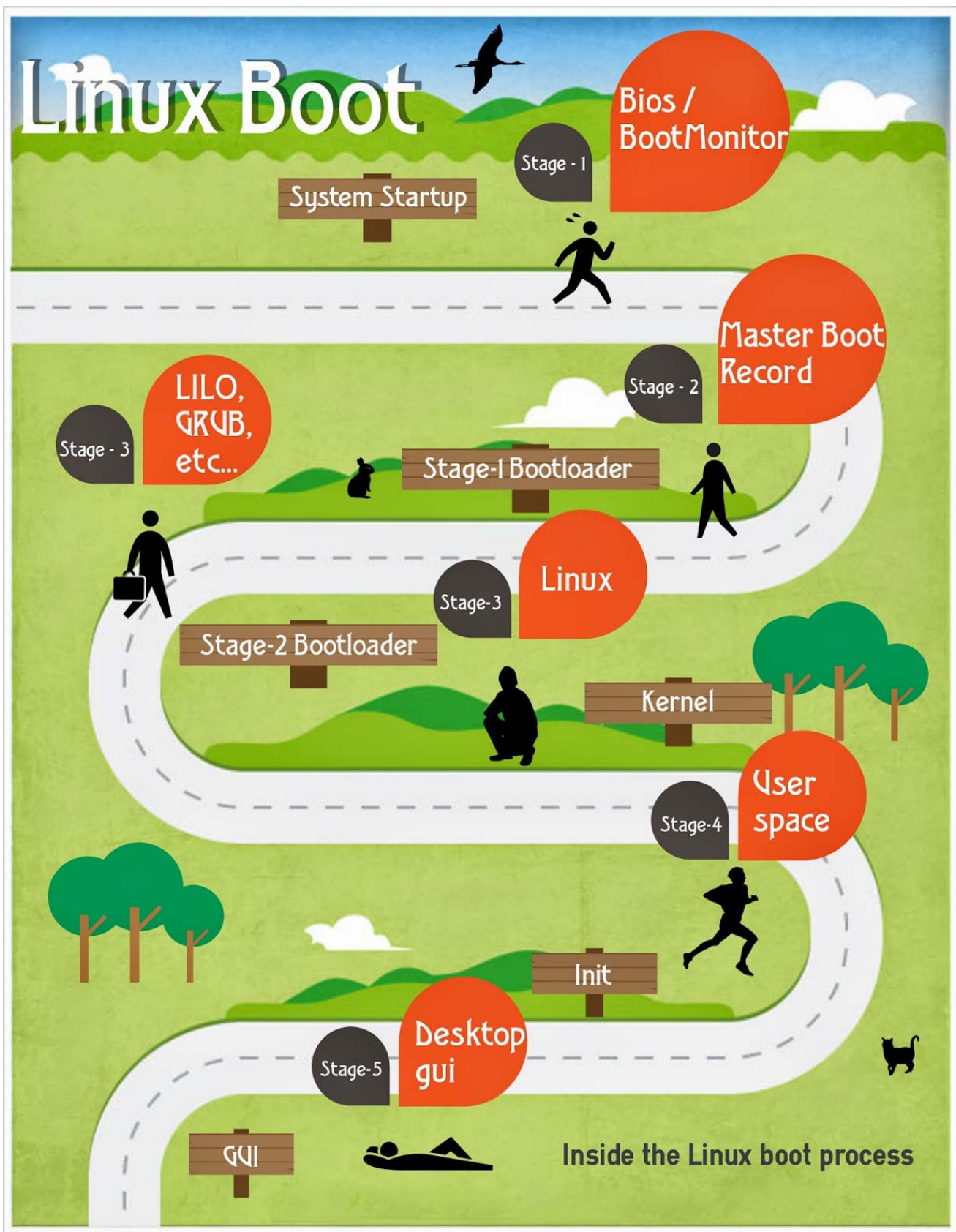
آنچه پیش رو دارید ویرایش **سوم** مقاله **فرایند Boot در لینوکس** است که به صورت رایگان به علاقه مندان لینوکس هدیه می گردد . در تهیه این مقاله از سر فصل های درسی گفته شده در دوره های LPic2 و RHCE استفاده شده و لازم می دانم از **مهندس مهدوی فر** به خاطر راهنمایی های مفیدشان و همچنین استاد **جعفر عقربی** تشکر کافی را داشته باشم. امید است این مطلب بتواند باعث ارتقاء دانش فنی کاربران لینوکس و متخصصین IT شود. اینجانب را از نظرات خود مطلع سازید.

موفق باشید

حسام الدین توحید

تیر 1395

5	▪ مقدمه‌ای بر فرایند Boot
6	▪ مرحله اول - Post
8	▪ انتخاب اولین دستگاه قابل Boot
8	▪ مرحله دوم - MBR
10	▪ مرحله سوم - Grub
11	▪ مرحله چهارم - Kernel
13	▪ مرحله پنجم - Init or systemd
16	▪ مرحله ششم - Run Levels
19	▪ روند ایجاد یک ترمینال



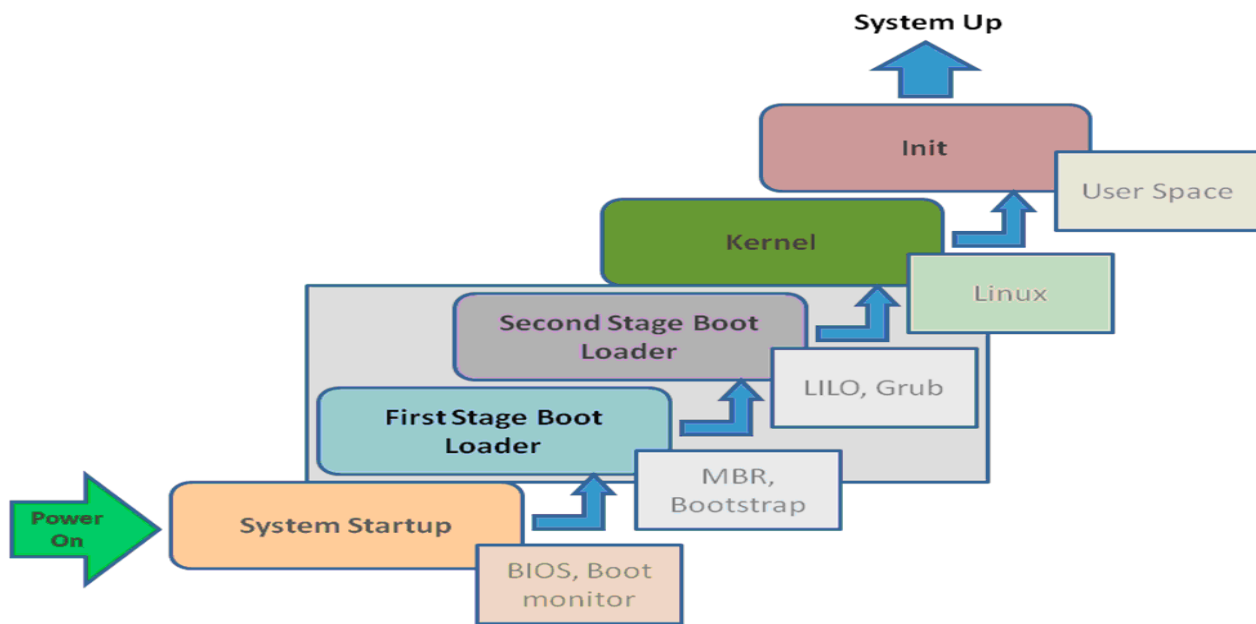
مقدمه ای بر فرآیند Boot

در این مقاله به این موضوع می پردازیم که سیستم عامل لینوکس چگونه Boot می شود و فرآیند Startup این سیستم عامل از زمان روشن شدن سیستم تا استفاده از خط فرمان داخلی آن چگونه انجام می شود. ما هر روز بارها کامپیوترمان را روشن کرده و برای چند ثانیه تا چند دقیقه شاهد بوت شدن سیستم عامل روی آن هستیم، اما تاکنون از خودمان سؤال کرده ایم در پس این بوت شدن چه اتفاقی می افتد تا سیستم عامل آماده اجرای کار می شود؟ کاربران لینوکس در هر بار روشن کردن کامپیوتر شاهد چندین ده خط هستند که پشت سر هم روی صفحه نمایش ظاهر شده و رد می شوند، بدون این که بدانند این خطوط چه می گویند. دانستن اینکه یک سیستم عامل چگونه Boot می شود هم می تواند به درک شما در برقراری ارتباط بین سخت افزار و نرم افزار و همچنین رفع مشکلات ناشی از Boot نشدن سیستم عامل بسیار کمک کند. در اینجا این فرآیند با جزئیات بیشتری در شش گام به شما معرفی می شود.

فرآیند کلی Boot سیستم عامل لینوکس شامل 6 مرحله اصلی است که همانطور که در تصویر زیر نیز مشاهده می کنید این مراحل به BIOS، MBR، Boot Loader، Kernel، Init و Runlevel تقسیم بندی می شود که در ادامه هر کدام از این مراحل، با ذکر جزئیات بیشتر توضیح داده می شوند.

لازم به ذکر است با ارائه سری هفت لینوکس ردهت، init جای خود را کاملاً به systemd داد و مفهوم runlevel هم با مفهوم جدیدی به نام Target جایگزین شد، لذا مفهوم و پروسه init در دنیای لینوکس رو به منسوخ شدن می رود و لازم است مدیران سیستم های لینوکسی به سمت فراگیری مفاهیم جانشین آن بروند. اما با توجه به اینکه سالها برای بیان پروسه بوت و بالا آمدن لینوکس از مفاهیم init و runlevel ها استفاده شده، در این مقاله هم از همین مفاهیم استفاده می شود اما در مقالات جداگانه ای با این دید که init جای خود را به systemd و بایوس هم جای خود را به مفهوم جدیدی به نام UEFI داده، توضیحات بیشتری داده خواهد شد.

نکته: تمامی مراحل بوت شدن تا وارد شدن به سیستم به همراه تمامی اتفاق های رخ داده (خطاها – هشدار ها و موفقیت ها در انجام فرایند) همگی در فایل های log در دایرکتوری /var/log/message/ و ثبت می شود.



مرحله اول - Post

اولین فاز فرآیند Boot شدن سیستم، فرآیند Startup است. زمانیکه شما دکمه Power سیستم را می‌زنید یا اینکه سیستم را Restart می‌کنید، قدرت برق به SMPS یا Switched Mode Power Supply می‌رسد که در این قسمت برق تبدیل به AC و یا DC می‌شود. برق DC به تمامی تجهیزات سیستم اعم از Motherboard، هارد دیسک‌ها، DVD-ROM، ماوس و کیبورد و ... داده می‌شود و آنها را روشن می‌کند. هوشمندترین قطعه کامپیوتر قطعا پردازنده یا همان CPU است، به محض اینکه برق به این دستگاه برسد این دستگاه شروع به انجام عملیات‌هایی می‌کند که در حافظه یا ROM سیستم وجود دارد. اولین دستوری که در این مرحله از طرف CPU صادر می‌شود این است که کنترل سیستم به دست BIOS سیستم داده شود، BIOS مخفف کلمات Basic Input/Output System است. دستور CPU به بایوس این است که عملیات POST یا Power On Self-Test را انجام دهد. در این عملیات تجهیزات ضروری سیستم برای فرآیند Boot کاملاً آزمایش می‌شوند.

بعد از اینکه کنترل به دست BIOS سیستم سپرده شد دو چیز اتفاق می‌افتد :

1. فرآیند آزمایش POST یا Power On Self-Test

2. انتخاب اولین دستگاه قابل Boot در سیستم



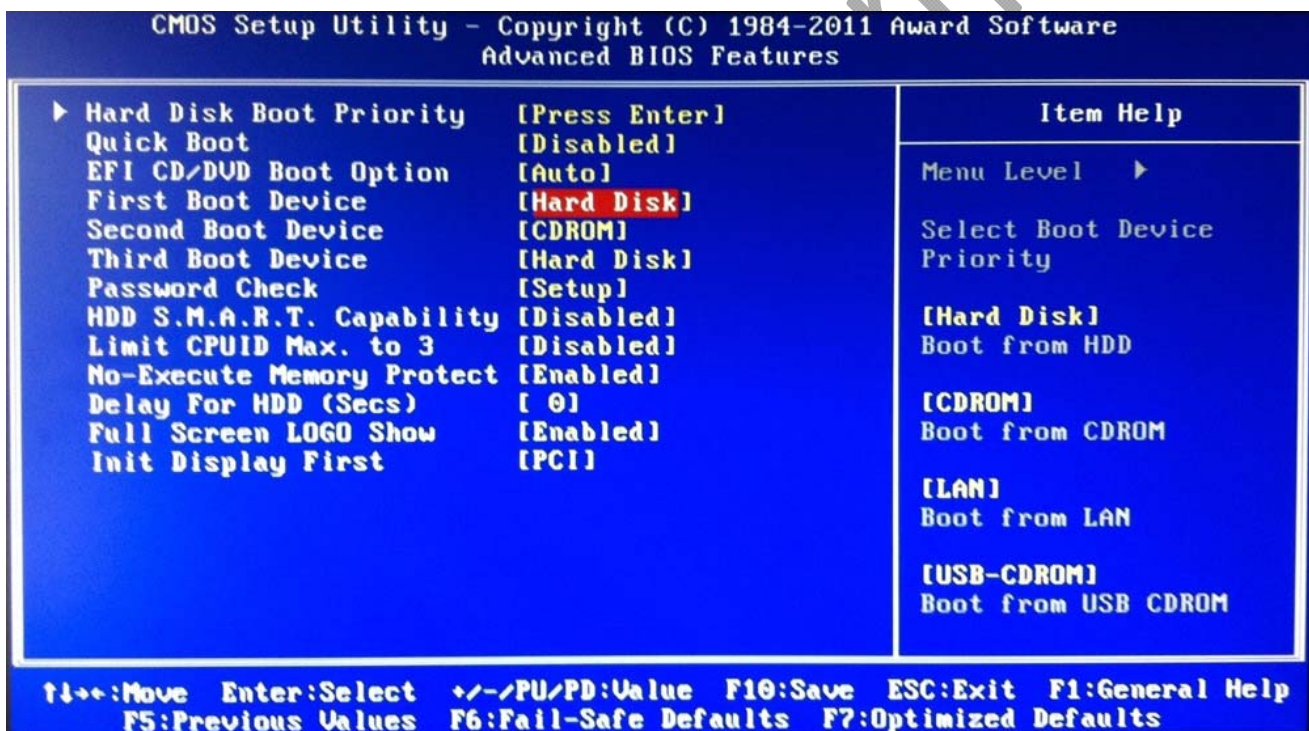
نکته: لینوکس به محض فعال شدن با یک فریم ورکی، CPU را آپدیت می کند.

عملیات POST یا Power On Self-Test یک سری فرآیند پردازشی است که در دسترس بودن سخت افزارهای سیستم را بررسی می کند. BIOS لیستی از سخت افزارهایی که در آخرین Boot موفق سیستم فعال بوده اند را همیشه در خود دارد، برای اینکه بررسی شود که یک سخت افزار برای انجام فرآیند Boot سیستم در دسترس است یا خیر برای هر کدام از این سخت افزارها که در لیست قرار دارند یک پالس الکتریکی فرستاده می شود.

اگر پالس الکتریکی به درستی پاسخ داده شود به معنی در دسترس بودن دستگاه مورد نظر و آماده بودن سخت افزار برای انجام فرآیند Boot است. اما اگر از سخت افزاری پالس دریافت نشود، سخت افزار مورد نظر به عنوان سخت افزار معیوب یا مشکل دار در نظر گرفته شده و فرض بر این گذاشته می شود که از سیستم جدا شده است. اینکار برای همه سخت افزارهای موجود در لیست BIOS انجام می شود و لیست جدید بر اساس آزمایش های جدید ایجاد شده و در حافظه BIOS برای فرآیند Boot بعدی استفاده خواهد شد.

انتخاب اولین دستگاه قابل Boot

زمانیکه عملیات‌های مربوط به POST کامل شدند، BIOS لیست دستگاه‌هایی که در دسترس هستند را خواهد داشت. در اینصورت BIOS به راحتی قادر خواهد بود اولین دستگاهی که در CMOS به عنوان Boot Device معرفی شده را انتخاب کند. در این مرحله، BIOS همیشه اولین دستگاه قابل Boot را انتخاب کرده و سپس کنترل سیستم را مجدداً به پردازنده یا CPU می‌سپارد. اگر فرض کنیم که اولین دستگاه قابل Boot بر روی سیستم پیدا نشود، بصورت خودکار BIOS به دنبال دومین دستگاه تعیین شده در لیست رفته و به همین ترتیب همه دستگاه‌های قابل Boot را آزمایش می‌کند تا به نتیجه مورد نظر برسد. اگر BIOS نتواند هیچ دستگاه قابل Boot ای بر روی سیستم پیدا کند به شما پیامی به شکل No boot device found به معنی عدم پیدا کردن دستگاه قابل Boot نمایش خواهد داد.

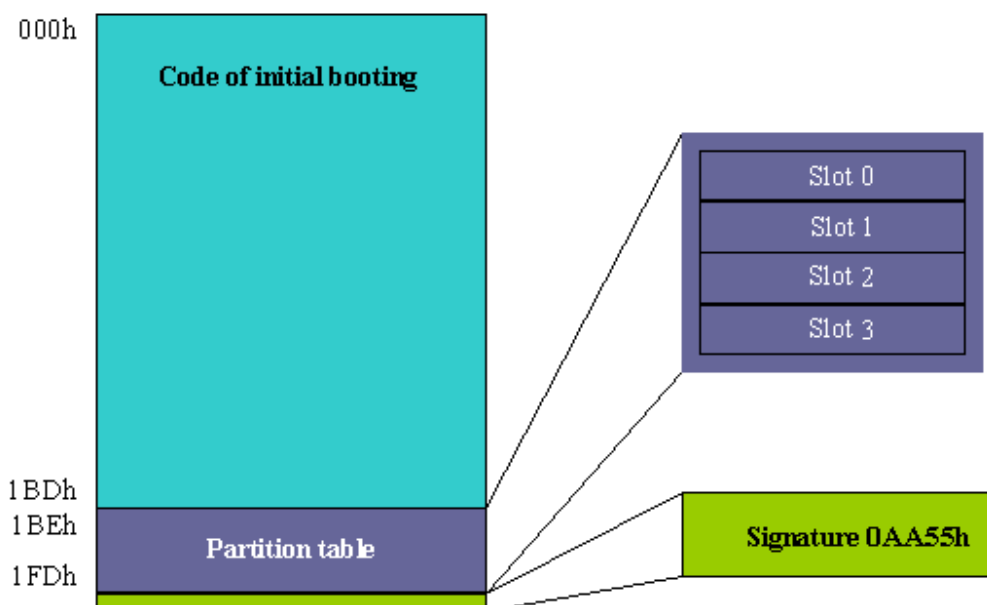


مرحله دوم - MBR

MBR مخفف Master Boot Record است. زمانیکه BIOS مجدداً اختیار یا کنترل سیستم را در اختیار CPU قرار می‌دهد، BIOS سعی می‌کند MBR را از اولین دستگاه قابل Boot خوانده و Load کند، در اینجا فرض را بر این گذاشته‌ایم که دستگاه Bootable ما هارد دیسک است. **MBR** یک قسمت کوچک از هارد دیسک‌ها است که اندازه‌ای برابر 512

بایت دارد، **تاکید میشود 512 بایت نه کیلو بایت**، این سکتور 512 بایتی از هارد دیسک یا MBR در اولین نقطه شروع نوشته شدن اطلاعات و یا در انتهای هارد دیسک بر اساس نوع سازنده آن قرار می‌گیرد. MBR جدولی است که داخل آن نام درایو اکتیو سیستم نوشته شده است. به پارتیشنی اکتیو گفته می‌شود که سیستم عامل در آن نصب شده است. در واقع می‌توان به MBR یک Boot Loader اولیه گفت که وظیفه اصلی آن فراخوان کردن Boot Loader ثانویه سیستم است چون Boot Loader در داخل فضای MBR نصب می‌باشد.

MBR بوت سکتور، حاوی برنامه اجرایی Boot Loader می‌باشد و پیش زمینه اجرا شدن GRUB را فراهم می‌کند و این امکان را می‌دهد که GRUB به داخل حافظه Load شده و باعث بالا آمدن Kernel شود. از 512 بایت فضایی که به MBR اختصاص یافته در حدود 446 بایت آن به Primary Boot Loader، 64 بایت آن به Partition Table و در نهایت 6 بایت آن برای اعتبارسنجی MBR یا MBR Validation و Timestamp آن اختصاص یافته است.



خوب تا اینجا کار، MBR به درستی Load شد اما MBR هیچگونه درکی از Kernel و نحوه Load کردن آن ندارد و نمی‌تواند مفهوم فایل سیستم را درک کند، در اینجا است که کار یک Boot Loader دیگر برای درایوهای فایل سیستم و Load کردن Kernel دیده می‌شود و این Secondary Boot Loader چیزی جز GRUB نیست و البته ممکن است در توزیعی از Boot Loader دیگری استفاده شود. مهمترین وظیفه MBR بارگزاری و اجرای GRUB یا Boot Loader پیش فرض سیستم است.

مرحله سوم - Grub

لینوکس یکی از پیشرفته‌ترین برنامه‌های بوت را دارد. در پروسه بوت، کارایی در کنار استفاده از کمترین منابع سخت افزاری و بالا بردن سرعت همراه با انعطاف پذیری منظور شده است. برنامه بوت لینوکس قابلیت تشخیص و سازگاری با دیگر برنامه های بوت سیستم عامل ها را دارد، به طوری که می تواند چندین سیستم عامل روی یک هارد دیسک را شناسایی و پیکربندی کند. اگر چندین سیستم عامل مانند دو توزیع لینوکسی و یک ویندوز را به صورت Multi Boot نصب کرده باشید حتما دیده‌اید که در مرحله‌ای از Boot سیستم از شما نوع سیستم عاملی یا کرنلی که می‌خواهید آن را Boot کنید سؤال می‌شود. به محض انتخاب هر کدام، گراب آن هسته را از هارد بر روی حافظه اصلی (RAM) قرار می‌دهد و تا زمانی که سیستم روشن است آن هسته بصورت دائمی در حافظه می‌ماند. اگر سیستم دارای بوت دوگانه باشد MBR حاوی **Primary Boot Loader Execute Program** است. این برنامه در لینوکس می تواند Lilo, Grub, Grub2 و یا هر Open Source Boot loader دیگری مانند OpenBoot، OpenBIOS، Syslinux، یا Grub باشد. یکی از پراستفاده‌ترین اینها Grub است.

```
GNU GRUB  version 0.95  (638K lower / 128960K upper memory)

root (hd0,0)
kernel /boot/vmlinuz-2.6.15-1-686 root=/dev/sda1 ro
initrd /boot/initrd.img-2.6.15-1-686
savedefault
boot
```

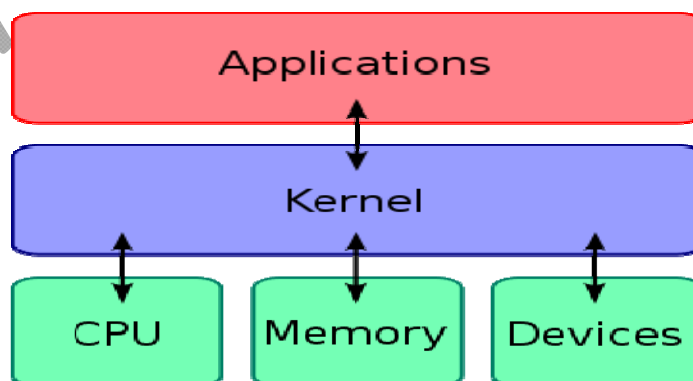
Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.

Boot loader در سه stage کار خود را انجام می دهد:

- **Stage 1:** در این مرحله MBR و فعالیتهای آن شناسایی می شوند.
 - **Stage 1/5:** در این مرحله فایل سیستم و درایورهای لازم برای بارگزاری شناسایی می گردند. این درایورها قبل از درایو Active و در سکتور 1 تا 63 هارد دیسک قرار دارند.
 - **Stage 2:** مرحله ای از بوت که Boot Loader اسامی سیستم عاملهایی را که روی هارد دیسک وجود دارند را detect کرده در صفحه Boot Menu نمایش می دهد تا آنها را بارگزاری کند.
- Grub محتوی فایل های Image مربوط به سیستم عامل ها و initrd است و وظیفه اصلی آن نیز اجرا و Load کردن همین Image ها است. تا اینجا کار هنوز هیچ قسمتی از سیستم عامل لینوکس فعال نشده است ، در واقع همه فرآیندی که مربوط به سیستم عامل می شود در مرحله بعدی انجام می شود که Grub کارهای خود را انجام داده و نوبت به Kernel سیستم عامل لینوکس می رسد که توسط Grub در حافظه RAM سیستم بارگذاری می شود . در قسمت بعدی در خصوص ادامه مراحل Load شدن Kernel و همچنین فرآیند Init توضیحاتی ارائه می شود. در مورد Grub در فصلی مجزا صحبت خواهد شد.

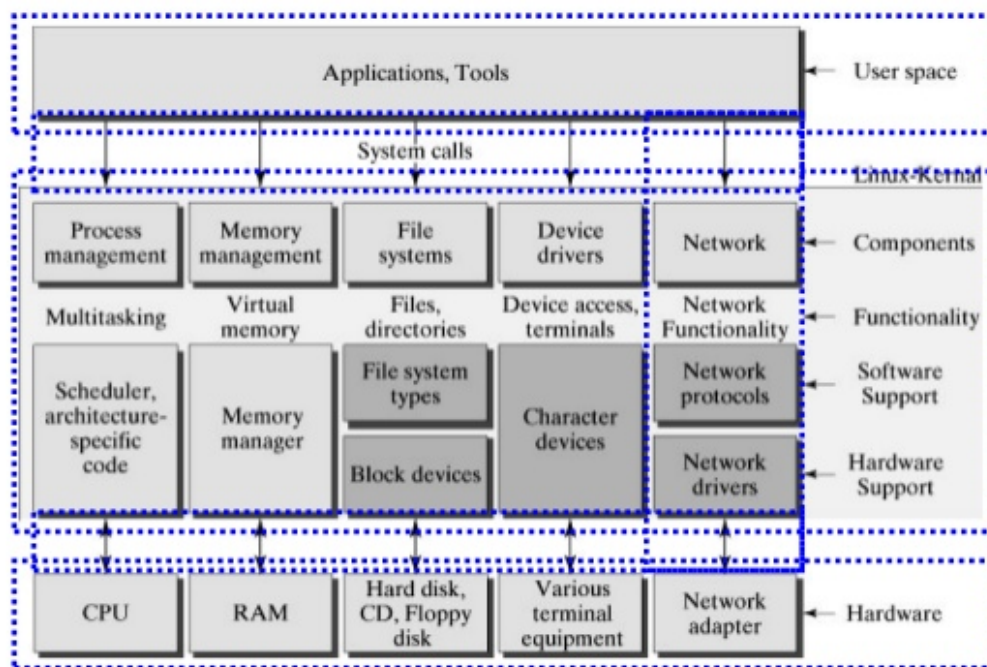
مرحله چهارم - Kernel

Kernel یا قلب تپنده سیستم عامل از هزاران ماژول نرم افزاری تشکیل شده است که تمامی فرآیندهای سیستم عامل را مدیریت می کند. هنگامی که Image هسته لینوکس درون حافظه قرار گرفته و اجرا می شود، لینوکس شروع به کار کرده و مدیریت بوت از دست GRUB خارج می شود. البته استفاده از لفظ « اجرا » در این مرحله کاملاً صحیح نیست، زیرا هسته عملاً اجرا نمی شود، بلکه از حالت فشرده خارج شده و چند سرویس محدود مانند سرویس های مرتبط با سخت افزار در حافظه اجرا می شوند. Image هسته لینوکس اگر کمتر از 512 کیلوبایت باشد به آن zImage و اگر بیشتر از این مقدار باشد، به آن bzImage گفته می شود. فایل های ایمج هسته با استفاده از ابزار zlib فشرده سازی و گشوده می شوند. همانطور که در شکل زیر می بینید ارتباط بین نرم افزارهای کاربردی و سخت افزار توسط کرنل سیستم عامل برقرار می شود.



در مرحله قبل Grub به جایی رسید که Image های مربوط به Kernel و initrd را به داخل حافظه RAM سیستم Load کرد و از این مرحله به بعد همه کارها بر عهده Kernel می باشد. Kernel به محض Load شدن تنظیمات مربوط به حافظه RAM اختصاص یافته به سیستم و همچنین پیکربندی کلیه سخت افزارهای سیستم را انجام می دهد سپس image مربوط به initrd را از حالت فشرده خارج می کند، در حالت پیش فرض این image فشرده و به شکل zlib یا zImage قرار گرفته است. بعد از اینکه از حالت فشرده خارج شد با mount کردن آن، درایورهای ضروری را Load می کند.

Linux Kernel Structure



لازم است در اینجا توضیح کوتاهی در مورد initrd داده شود.

در لینوکس های قدیمی درایورهای لازم برای شناسایی Storage ها و بعضی از قطعات در سیستم عامل موجود نبود به همین خاطر در طی مراحل بوت، هارد دیسک توسط Boot Loader شناسایی نمی شد. لذا متخصصین این حوزه سازوکاری فراهم کردند تا قبل از اینکه Kernel خوانده و Load شود، در Ram سیستم یک فضای کوچکی از پارتیشن root تشکیل شود تا بتوان در این فضا درایورهای لازم برای شناسایی قطعات که شناسایی نمی شوند را بارگذاری کرد. به این فضای کوچک Ram Disk گفته می شود. این فضا در لینوکس های پیشرفته حال حاضر جهت احتیاط وجود دارد ولی دیگر از آن استفاده ای نمی شود. محتویات Ram Disk را با دستور lsinitrd می توان دید. پروسه initrd امکان بوت هسته لینوکس را بدون نیاز به هیچ دسترسی فیزیکی به هارد دیسک و بالا بردن سرعت عمل بیشتر هسته فراهم می کند. هنگامی که هسته لینوکس به طور کامل اجرا می شود، این پروسه اصطلاحاً unmount شده و سیستم فایل حقیقی به کرنل متصل می گردد. در حقیقت پروسه initrd امکان ایجاد یک هسته لینوکس کوچک و فشرده شده را فراهم می کند که به این وسیله

میتوان لینوکس را به راحتی اجرا یا روی سیستم‌های دیگر بدون وابستگی به هارد دیسک منتقل کرد. Load و Unload کردن ماژول‌های Kernel لینوکس با استفاده از برنامه‌هایی به نام insmod و rmmod انجام می‌شود که در initrd image قرار گرفته‌اند.

در اینجا Kernel به دنبال هارد دیسک می‌رود و بررسی می‌کند که در حالت LVM قرار دارند و یا اینکه RAID شده‌اند. سپس initrd را Unmount و فضای اختصاص یافته توسط این Image را نیز خالی می‌کند. Kernel پارتیشن ریشه یا root partition ای که در فایل grub.conf وجود دارد را در حالت فقط خواندنی یا read only به سیستم Mount کرده و در نهایت فرآیند init یا systemd را اجرا می‌کند. توجه کنید که Kernel تا زمانیکه سیستم شما خاموش نشده باشد بر روی حافظه RAM شما باقی می‌ماند.

کرنل بعد از اینکه کارهایش را انجام داد می‌آید پروسه‌ای را واسطه قرار داده و به آن اختیار می‌دهد تا همه کارها را مدیریت کند، این پروسه init یا systemd می‌باشد.

مرحله پنجم - Init or systemd

اگر دقت کرده باشید بعد از اینکه از صفحه انتخاب سیستم عامل Grub عبور کردید به یکباره یک سری خطوط به شما نمایش داده می‌شود و در کنار آنها بعضاً کلماتی مثل OK و Failed را مشاهده می‌کنید که به سرعت از جلوی چشم شما عبور می‌کند (شکل زیر)، سیستم در حال Start کردن سرویس‌های لازم جهت اجرای سیستم عامل می‌باشد.

```
295
audit(1216470015.968:3): policy loaded audit=4294967295 ses=4294967295
INIT: version 2.86 booting
      Welcome to Red Hat Enterprise Linux Server
      Press 'I' to enter interactive startup.
Setting clock (utc): Sat Jul 19 05:20:22 MST 2008          [ OK ]
Starting udev:                                           [ OK ]
Loading default keymap (us):                            [ OK ]
Setting hostname rhce-prep.example.com:                 [ OK ]
No devices found
Setting up Logical Volume Management:
  No volume groups found                                [ OK ]

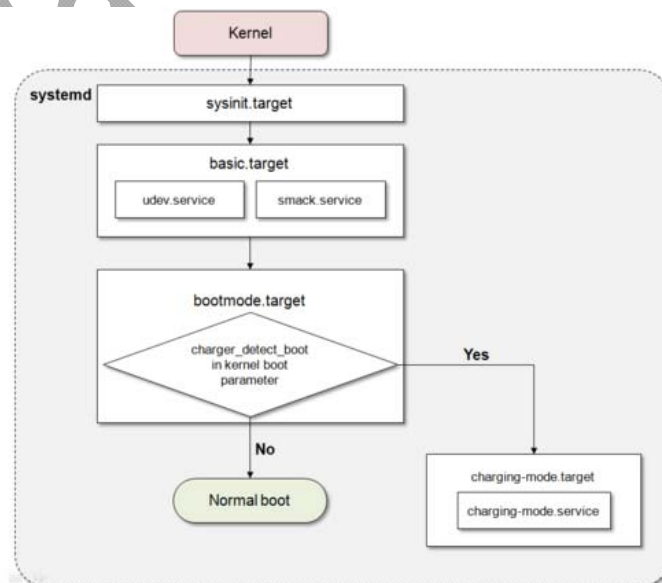
Checking filesystems
/: clean, 4871/263232 files, 72321/263056 blocks
/home: clean, 117/130560 files, 27384/522080 blocks
/var: clean, 1165/130560 files, 65117/522080 blocks
/dev/md0: clean, 12/883872 files, 45604/883456 blocks
/usr: clean, 81733/524288 files, 427747/524120 blocks
/boot: clean, 33/66264 files, 24068/265040 blocks        [ OK ]

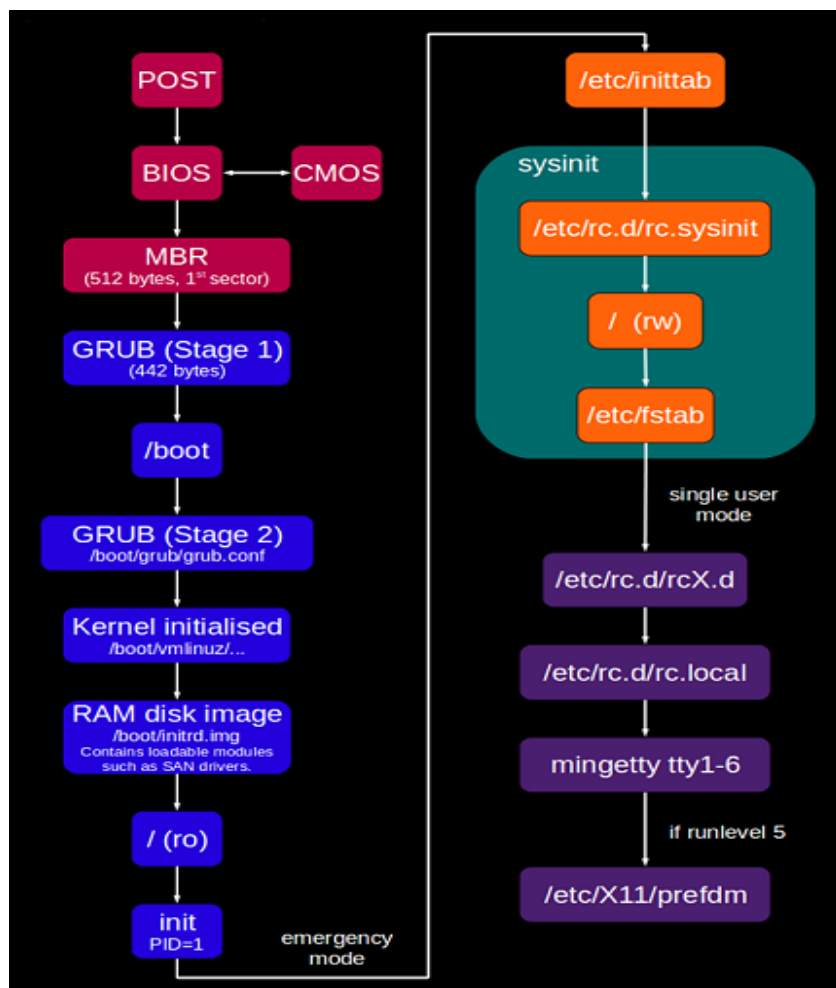
Remounting root filesystem in read-write mode:          [ OK ]
Mounting local filesystems:                             [ OK ]
Enabling local filesystem quotas:                       [ OK ]
```

به محض اینکه Kernel در Ram بارگزاری شد یک فرآیند یا Process به نام init یا systemd توسط Kernel اجرا می‌شود که اولین Process اجرا شده توسط Kernel است. init مخفف کلمه initialization بوده که در ترجمه فارسی به معنی مقدار دهی اولیه است. init پردازشی است که همه پردازش‌های مورد نیاز سیستم عامل را فراخوانی و اجرا می‌کند. این پروسس Parent تمام پروسس‌های سیستم می‌باشد.

زمانیکه init اجرا شد به عنوان فرآیند والد یا Parent Process در سیستم عامل لینوکس در نظر گرفته می‌شود. اولین کاری که init انجام می‌دهد این است که محتویات فایل تنظیمات خودش یا initialization file ای که در مسیر **/etc/inittab** وجود دارد را بخواند. محتویات این فایل به init می‌گوید که یک اسکریپت تنظیمات اولیه محیطی یا environment configuration script را اجرا کند که در این اسکریپت تعیین مسیرها یا path ها ، فرآیند Swapping ، بررسی فایل سیستم و ... انجام می‌شود. تقریباً می‌توان گفت اجرای این اسکریپت هر چیزی که سیستم شما نیاز دارد تا فرآیند مقداردهی اولیه یا همان initialization را انجام دهد را شامل می‌شود ، حتی تعیین کردن ساعت سیستم، پورتهای سریال و ... همه در این مرحله انجام می‌شود. در ادامه این فایل به init اعلام می‌کند که سیستم قرار است چگونه و در چه run level پیکربندی شود .

run level در واقع پیکربندی فرآیندها یا پردازش‌های موجود در سیستم است. بعد از اینکه run level پیشفرض برای سیستم در نظر گرفته شد ، init با توجه به محتویات دایرکتوری rc ای که به run level مربوطه اختصاص یافته است تمامی پردازش‌ها یا Process های پس زمینه‌ای که برای اجرا شدن سیستم، مورد نیاز هستند را اجرا می‌کند. Init هر کدام از اسکریپت‌های kill را که با حرف K شروع می‌شوند را با یک پارامتر stop و سپس تمامی اسکریپت‌های Start که با S شروع می‌شوند را اجرا و سرویس‌ها و برنامه‌های کاربردی که در run level مربوطه وجود خواهند داشت را اجرایی خواهد کرد. نکته جالب در خصوص این اسکریپت‌ها این است که شما می‌توانید آنها را بعد از وارد شدن به سیستم بصورت دستی نیز stop و یا start کنید ، اینکار براحتی با استفاده از دستور service در init قابل اجرا است .



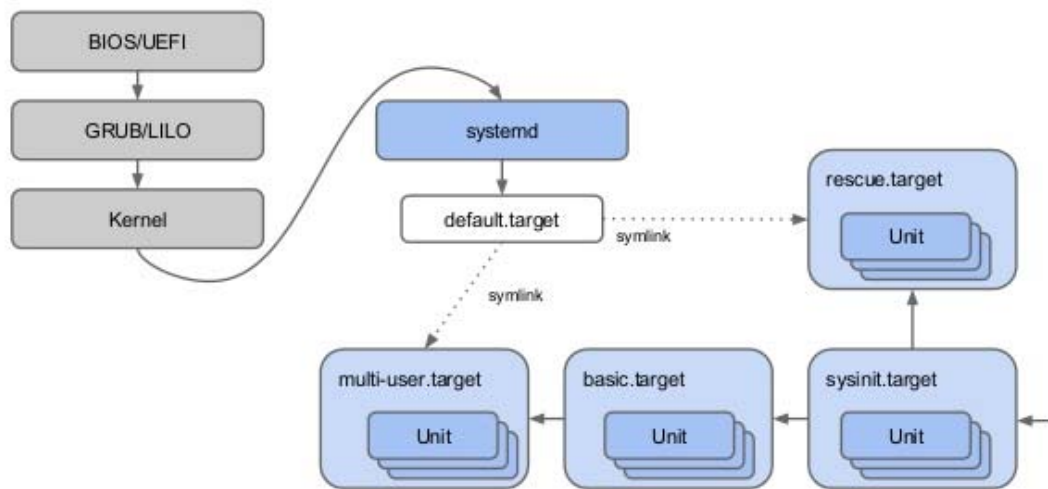


توجه کنید که در زمان startup سیستم معمولا دو اسکریپت rc3.d و rc2.d اجرا می شوند. در این حالت هیچ سرویسی در حالت stopped قرار نمی گیرد یا حداقل می توان گفت بصورت دائمی در حالت stopped قرار نمی گیرد. هیچکدام از اسکریپت هایی که در پوشه `/etc/rc<x>.d/` قرار گرفته اند سرویس ها را stop یا start نمی کنند، بلکه تمامی فایل هایی که در پوشه `/etc/rc<x>.d/` قرار گرفته اند به عنوان یک لینک به اسکریپت هایی که در پوشه `/etc/init.d/` قرار دارند اشاره می کنند، به اینگونه لینک ها symbolic link گفته می شود، یک symbolic link چیزی بیشتر از یک فایل که به یک فایل دیگر اشاره می کند نیست و زمانی استفاده می شود که شما می خواهید بدون ایجاد کردن و یا حذف کردن یک فایل اسکریپت آن را اجرا و سرویس ها را start و یا stop کنید. در نهایت در این قسمت اسکریپت های مربوط به run level مورد نظرتان انتخاب و اجرا خواهند شد و سیستم به سراغ مرحله بعدی می رود.

تا قبل از ارائه systemD مخزن لینوکس دایرکتوری init.d بود و تمامی فایل هایی که در دایرکتوریهای rc قرار داشتند به این دایرکتوری لینک می شدند. به محض اینکه یک سرویس در لینوکس نصب می شود یک اثر از خودش در init.d به جا می گذارد. در نسخه های قدیمی لینوکس، init فقط یک فایل پیکربندی داشت که گاهی به چند صد خط می رسید. یعنی تمام فایل هایی که در دایرکتوریهای rc قرار دارد در یک فایل کانفیگ بزرگ جمع شده بودند که این خود یک ضعف و

آسیب پذیری به شمار می‌رود، چون اگر برای این فایل مشکلی پیش می‌آمد سیستم up نمی‌شد. لذا در نسخه‌های بعدی، آن فایل بزرگ چند صد خطی به تعداد زیادی فایل تبدیل شد تا مدیریت و پایداری سیستم تضمین بشود. init اولین نرم افزاری است که در user mode کار کرده و کلاً به صورت event base عمل می‌کند یعنی یک چیزی باید اجرا شود تا مورد بعدی به مرحله اجرایی شدن برسد. این سریالی عمل کردن init در اجرای سرویس‌ها یکی از ضعف آن محسوب می‌شود.

systemd boot process



مرحله ششم - Run Level

مهم ترین وظیفه پروسه init تعیین نوع اجرای لینوکس یا تعیین سطح اجرایی کاربری لینوکس (RunLevel) است که شما آن را روی صفحه نمایش می‌بینید. همانطور که در مرحله قبل مشاهده شد در فرآیند init بر اساس تنظیمات فایل inittab یک run level برای سیستم انتخاب می‌شود که بر حسب run level انتخابی یک سری اسکریپت از پوشه‌های مرتبط با آن اجرا می‌شوند.

startup script های مربوط به init در پوشه **/etc/rc.d/** قرار گرفته‌اند اما اسکریپت‌هایی که برای run level ها استفاده می‌شوند در subdirectory هایی به شکل **/etc/rc.d/rc0.d/ تا /etc/rc.d/rc6.d/** بر اساس انتخاب init از بین run level های 0 تا 6 اجرا می‌شوند. در آخر نیز init هر چیزی که در پوشه **/etc/rc.d/rc.local/** پیدا کند را فارق از اینکه در چه run level ای قرار دارد اجرا خواهد کرد. در زیر، پوشه‌های مربوط به انواع run level script لیست شده است که متناسب با run level انتخاب شده در init اجرا خواهند شد.

Runlevel Directory

/etc/rc.d/rc0.d
 /etc/rc.d/rc1.d
 /etc/rc.d/rc2.d
 /etc/rc.d/rc3.d
 /etc/rc.d/rc4.d
 /etc/rc.d/rc5.d
 /etc/rc.d/rc6.d

به طور کلی، لینوکس می تواند در هفت سطح اجرا شود. این سطوح به شرح زیر می باشند:

- سطح 0: خاموش شدن سیستم
- سطح 1: اجرای لینوکس در حالت تک کاربره
- سطح 2: اجرای لینوکس در حالت چند کاربره بدون دسترسی به شبکه
- سطح 3: اجرای لینوکس در حالت چند کاربره با دسترسی به شبکه
- سطح 4: غیر قابل استفاده
- سطح 5: اجرای لینوکس در حالت چند کاربره با شبکه و محیط گرافیکی دسکتاپ
- سطح 6: ریستارت سیستم

تمامی سیستم عامل های خانواده Unix توانایی این را دارند که با پیکربندی های پردازشی مختلف اجرا شوند به طور مثال ما می توانیم به سیستم عامل لینوکس بگوییم که در حالت تک کاربره یا single user mode اجرا شود لذا در این حالت init سیستم را در run level 1 بالا می آورد و در این حالت فقط مدیر سیستم می تواند به سیستم متصل شود. معمولاً زمانی از چنین run level ای استفاده می شود که می خواهیم وظایف نگهداری سیستم یا maintenance task های سیستم را بدون ریسک تخریب سیستم یا داده های کاربران انجام دهیم. طبیعتاً در چنین run level ای ما نمی خواهیم هیچگونه سرویسی به کاربران ارائه شود و سرویس های کاربری معمولاً در این run level غیر فعال یا Disable هستند.

Run Level دیگری که استفاده می شود به نام reboot run level شناخته می شود یا run level 6 که وظیفه shutdown کردن همه سرویس ها و دستورالعمل های وابسته به آن و همچنین restart کردن سیستم را به عهده دارد. به این موضوع دقت کنید که اسکریپتی که در پوشه rc.local وجود دارد در همه توزیع های لینوکس ممکن است وجود نداشته باشد.

در نهایت اگر همه چیز به خوبی پیش برود شما می توانید صفحه login مربوط به سیستم عامل لینوکس را مشاهده کنید. معمولاً لینوکس به صورت پیش فرض بر روی run level سه یا پنج تنظیم شده است، اما می توان این وضعیت را تغییر داد. برای تعیین نوع run level لینوکس به فایل پیکربندی /etc/inittab/ مراجعه کنید.

پس از اجرای کامل هسته لینوکس، برنامه سطح کاربری یا همان محیط گرافیکی اجرا می‌شود. این برنامه اولین نرم افزاری است که به وسیله کامپایلر و توابع کتابخانه ای زبان C اجرا می‌شود. در محیط گرافیکی دسکتاپ اولین پروسه **/sbin/init/** است که اجرا می‌شود.

اگر طبق run level مربوطه محیط گرافیکی نداشتیم یک صفحه مشکی و کامندی جهت Login به سیستم نشان داده می‌شود. در اینجا باید نام کاربری و پسورد خود را وارد کنید. اگر نام کاربری و پسوردتان درست باشد با ایجاد یک ترمینال توسط برنامه **mingetty** که یکی از 6 ترمینال tty1 تا tty6 می‌باشد، یک جلسه یا Session به نام یوزر لاگین کننده اختصاص داده می‌شود. لیست run level های مختلف در سیستم عامل لینوکس را می‌توانید در شکل زیر با مشخصات آنها مشاهده کنید. اما روند ایجاد یک ترمینال چگونه است؟

Run levels

init 0 : Shutdown the system or halt

init 1 : Single-user mode.

init 2 : Multi-user mode with out networking support.

init 3 : Multi-user mode with networking support.

init 4 : Unused.

init 5 : Graphical user interface.

init 6 : Reboot the system.

روند ایجاد یک ترمینال

Shell یا پوسته، ابزاری است که دستورها را از کاربر گرفته آنها را تحلیل کرده یا یک خروجی مناسب یا یک خطا در صورت مشکل داشتن وردی، در صفحه نمایش نشان می‌دهد. پروسه‌ای به نام fork وجود دارد که یک نمونه از خود فرایند را عیناً ایجاد می‌کند. وظیفه fork ایجاد پروسس child از فرایند parent می‌باشد. (از فرایند اصلی یا والد یک فرایند فرزند ایجاد می‌کند) پس از آخرین گام بالا دستور `/sbin/mingetty/` یک ترمینال را ایجاد می‌کند. Shell و Terminal دو چیز متفاوت هستند. در اصل پوسته یا شل کدهایی هستند که دستورهای ورودی از طریق ترمینال یا کنسول را تحلیل می‌کند اما ترمینال یک ابزاری برای کار با شل است. در لینوکس تعدادی پوسته وجود دارد مانند `bash, sh, csh, ksh, zsh` لیست نام این شل‌ها در لینوکس در فایل `shells` که با دستور زیر قابل مشاهده است قرار دارد.

#less /etc/shells

اما یک شل (از هر نوعی که باشد) چگونه در اختیار شما قرار می‌گیرد. همانطور که گفتیم شل بعنوان تحلیل‌گر دستورهای ورودی و ترمینال نیز بعنوان رابطی برای اتصال به شل استفاده می‌شود. بصورت پیش فرض `bash` بعنوان پوسته پیش فرض لینوکس است. به محض `login` شدن و ایجاد یک ترمینال توسط **mingetty** مثل `ttty1`، فرایند `init` نیز یک فرایند دیگر به نام `bash` را می‌سازد که پوسته `bash` را فعال می‌کند. بطور خلاصه می‌توان گفت که :

پس از مرحله ششم از مراحل بوت شدن، صفحه `login` نشان داده می‌شود و فرایند `init` نیز شل `bash` را از مسیر `/bin/bash/` ایجاد و کنترل می‌کند (توسط عمل `fork` یک نمونه از `bash` را می‌سازد) سپس در صورت درست وارد کردن نام کاربری و پسورد، برنامه **mingetty** یک ترمینال مانند `ttty2` را در اختیار شما قرار می‌دهد. توجه کنید که خود فرایند **mingetty** بعنوان فرزند `bash` و خود `bash` نیز بعنوان فرزند `init` می‌باشد.

اگر بخواهیم از ترمینال خارج شویم یا بهتر اینکه بخواهیم جلسه ایجاد شده را از بین ببریم باید از دستور `logout` یا `exit` استفاده کنیم. به محض اجرای این دستورها جلسه کاری تمام می‌شود و دوباره صفحه `login` نشان داده می‌شود.

نکته: در حین بوت شدن ممکن است اعمال دیگری مانند اجرای دستور `fsck` یا اعمال دیگری انجام شود ولی 6 مرحله بالا اساس راه اندازی یک سیستم عامل لینوکسی است.