

فهرست مطالب

مقدمه

بخش اول

تعاریف اولیه

۱- BIOS (Basic Input/Output System)

۲- بارگذار راه انداز (Boot Loader)

۳- سکتور راه انداز (Boot Sector)

۴- سکتور راه انداز اصلی (Master Boot Record)

۵- پارتیشن پایه (Primary Partition)

۶- پارتیشن منطقی (Logical Partition)

۷- پارتیشن فعال (Active Partition)

قواعد نامگذاری دیسکهای سخت و پارتیشن ها

بخش دوم

تاریخچه گراب (GRUB:GRand Unified Bootloader)

امکانات GRUB

نحوه تهیه و نصب GRUB

دنبال کردن رویه راه اندازی سیستم

GRUB به عنوان بارگذار راه انداز سیستم

بررسی ساختار فایل menu.lst

یک مثال از فایل menu.lst

ویرایش اجزاء منوی راه اندازی در طول رویه بالا آمدن سیستم

استفاده از Wild Card ها جهت انتخاب هسته در زمان راه اندازی

فایل device.map

تنظیم یک گذرواژه برای بارگذار راه انداز

مراحل تهیه یک فلاپی یا CD راه انداز GRUB

پرسشهای متداول GRUB

(۱) مشکل نصب Windows پس از نصب GNU/Linux

(۲) GRUB و سیستم فایل XFS

(۳) GRUB و سیستم فایل JFS

(۴) GRUB پیغام خطای "GRUB Geom Error" را نمایش می دهد

(۵) سیستم هایی که همزمان دارای دیسک IDE و SCSI هستند ، راه اندازی نمی شوند

(۶) راه اندازی Windows از دیسک سخت دوم

(۷) راه اندازی Windows در صورت غیرقابل دسترس بودن گنو/لینوکس

پیکربندی بارگذار راه انداز بوسیله ابزار گرافیکی Yast

تشریح اجزاء پیکربندی

حذف بارگذار راه انداز گنو/لینوکس

بخش سوم

مقدمه ای بر LILO

اجزا تشکیل دهنده LILO

۱- نصب کننده نقشه (map installer)

۲- بارگذار راه انداز سیستم (Boot loader)

۳- فایل‌های متفرقه (Various files)

فایل lilo.conf و فرمان lilo

اجرای برنامه LILO

اجرای برنامه LILO

ساخت یک دیسک راه انداز بدون استفاده از بارگذار راه انداز

پیامهای خطای LILO

حالت اول : عدم نمایش هیچیک از حروف LILO

حالت دوم : نمایش حرف L

حالت سوم : نمایش حروف LI

حالت چهارم : نمایش حروف LIL

حالت پنجم : نمایش LIL?

حالت ششم : نمایش LIL –

حالت هفتم : نمایش کلمه LILO

راه حل های موجود

پرسشهای متداول LILO

اطلاعات پایه و نصب استاندارد

در کجا باید LILO را نصب کرد؟

چگونه می توان درایورهای هارد IDE را پیکربندی کرد؟

نحوه عملکرد در زمان راه اندازی

چگونگی Unistsll کردن LILO

پیکربندی ساده lilo.conf

روش نصب Windows ازمنوی LILO

استفاده از LILO زمانی که BIOS قادر به دیدن پارتیشن ریشه نمی باشد

مرور دقیق تر بازیابی یک سیستم آسیب دیده توسط یک دیسک زنده

مراجع



مقدمه

در این کتاب قصد داریم به بررسی ساختار دیسک های سخت و بارگذاری را ه انداز موجود در لینوکس شامل GRUB و LILO پرداخته و نحوه نصب و را ه اندازی و اشکال زدایی آنها را مطرح نماییم. همچنین آشنایی با سایر اجزا تشکیل دهنده آنها و بررسی نحوه عملکرد این راه اندازها از دیگر مباحث اصلی این کتاب می باشند. عناوین و فصول کتاب نیز به نحوی طبقه بندی شده اند که مطالعه آنها برای سایر اشخاصی که زمینه قبلی در رابطه با مفاهیم راه انداز سیستم و ساختار دیسک ها ندارند ، مشکلی ایجاد نکرده و به راحتی قابل درک و استفاده خواهند بود.

مباحث کتاب در سه بخش که هر یک شامل فصل هایی مجزا می باشند ، گردآوری شده است. در بخش اول مفاهیم اولیه رویه راه اندازی سیستم و قواعد نامگذاری دیسک های سخت و پارتیشن های آنها در لینوکس بررسی می شود. سپس در بخش دوم کتاب سایر اجزا و پارامترهای راه انداز GRUB مطرح شده و در انتها به بررسی مشکلات احتمالی آن پرداخته می شود. در آخرین بخش هم که بخش سوم کتاب می باشد راه انداز LILO همانند GRUB مورد بررسی قرار گرفته و مباحثی چون نحوه تهیه ، پیکربندی اجزاء و اشکال زدایی آن مورد بررسی قرار می گیرد.

در انتهای هر بخش نیز منابع بکاررفته در آن معرفی شده اند تا علاقمندان در صورت تمایل جهت کسب اطلاعات بیشتر به آن مراجعه نمایند. امید است که این کتاب به عنوان یک مرجع فارسی بتواند اطلاعات لازم و کافی در ارتباط با بارگذارهای راه انداز سیستم عامل گنو/ لینوکس و مشکلاتی را که اغلب کاربران در زمان نصب گنو/ لینوکس در کنار سایر سیستم عامل ها با آن مواجه هستند را در اختیار آنها قرار داده و برای سایر کاربران و علاقمندان این عرصه مفید واقع شود.

بخش اول

تعاریف اولیه

قبل از آشنایی با نرم افزارهای GRUB و LILO و بررسی چگونگی عملکرد آنها بهتر است با برخی از مفاهیم که تا انتهای کتاب با آنها برخورد خواهیم داشت آشنایی لازم را پیدا کرده و سپس به مطرح کردن سایر مطالب در این مقوله بپردازیم . تعاریف موجود در ارتباط با هر یک از مفاهیم عنوان شده ، دید بازتری را نسبت به نحوه عملکرد سیستم و سایر اجزاء آن خصوصاً راه اندازهای سیستم ، در اختیار ما قرار می دهد.

۱- BIOS (Basic Input/Output System) :

BIOS عبارتست از برنامه ای که در داخل یک حافظه فقط خواندنی (ROM) برنامه ریزی شده و بر روی بورد مادریک کامپیوتر قرار می گیرد. وظیفه اصلی BIOS در زمان روشن شدن سیستم است، که باید در این زمان تست اولیه سیستم جهت راه اندازی را انجام دهد که به این مرحله POST (Power On Self Test) می گویند. در این قسمت BIOS وجود حداقل سخت افزار لازم جهت راه اندازی سیستم را که شامل صفحه کلید، صفحه نمایش ، RAM و بورد VGA می باشند، تشخیص می دهد. در نهایت BIOS باید یک آدرس مشخص از دیسک برنامه راه انداز سیستم را به داخل حافظه بارگیری نموده و آغازگر رویه راه اندازی سیستم باشد.

۲- بارگذار راه انداز (Boot Loader) :

برنامه ای است که در سکتور آغازین دیسک سخت سیستم قرار می گیرد. در خلال رویه راه اندازی سیستم ، پس از اینکه BIOS تست قطعات سخت افزاری سیستم را انجام داد، کنترل اجرای سیستم را به محتویات اولین سکتور دیسک واگذار می کند و برنامه داخل آنرا اجرا می نماید. این برنامه که وظیفه انتقال کنترل سیستم به سیستم عامل را برعهده دارد، بارگذار راه انداز نامیده می شود که از دو بخش تشکیل شده است و در رویه راه اندازی سیستم نقش اساسی ایفا می کند. از انواع بارگذارهای راه انداز موجود می توان به بارگذارهای GRUB و LILO که مخصوص گنو/ لینوکس بوده و NTLDR که مخصوص ویندوز است اشاره کرد.

۳- سکتور راه انداز (Boot Sector) :

به ناحیه ای از دیسک سخت، فلاپی دیسک، یا هروسیله ذخیره سازی اطلاعات اطلاق می شود که معمولاً ۵۱۲ بایت ابتدای آن می باشد. هرپارتیشن دارای یک سکتور راه انداز مخصوص خود بوده و تمام دیسک سخت نیز دارای یک سکتور راه انداز اصلی یا MBR می باشد. محتویات سکتور راه انداز به داخل حافظه بارگیری شده و به عنوان جزئی از بارگذار راه انداز سیستم اجرا می شود. در واقع محل قرار گرفتن

بارگذار راه انداز سیستم سکتور راه انداز می باشد و به محض روشن شدن سیستم ، اولین مکانی که هارد دیسک به سراغ آن رفته و اطلاعات مورد نیاز جهت راه اندازی سیستم را از آنجا بارگیری می کند، سکتور راه انداز یا سکتور صفر سیستم می باشد.

۴- سکتور راه انداز اصلی (Master Boot Record) :

اصولاً دیسک های پارتیشن بندی شده تنها دارای یک MBR می باشند ، در حالیکه در دیسک های پارتیشن بندی نشده به جای MBR از یک سکتور راه انداز که اولین سکتور دیسک نیز می باشد استفاده می شود. در هر دو شکل پس از خواندن محتویات اولین سکتور به داخل حافظه توسط BIOS، کنترل سیستم به برنامه موجود در این سکتور واگذار می شود. سکتور راه انداز اصلی سیستم یا MBR متشکل از سه بخش اساسی می باشد:

بخش اول آن بارگذار راه انداز سیستم بوده و شامل ۴۴۸ بیت می باشد. بارگذاری که در این ناحیه از اولین سکتور دیسک قرار می گیرد، می تواند GRUB , LILO و یا NTLDR باشد که همانطور که قبلاً گفته شد وظیفه بارگذاری سیستم عامل به داخل حافظه اصلی کامپیوتر را داراست.

دومین ناحیه از MBR حاوی جدول پارتیشن های دیسک سخت سیستم می باشد که مشتمل بر ۴ بخش ۱۶ بیتی است و جمعاً یک فضای ۶۴ بیتی را شامل می گردد. هریک از قسمت های ۱۶ بیتی موجود در این ناحیه ، جهت نگهداری مشخصات یکی از پارتیشن های پایه سیستم بکار می رود که از حجم این ناحیه می توان به این مطلب پی برد که حداکثر ۴ پارتیشن پایه می توان در یک دیسک ایجاد کرد.

سومین و در واقع آخرین بخش MBR ، یک فضای ۲ بیتی است که حاوی عدد AA55 در مبنای شانزده می باشد. این مقدار مشخص کننده انتهای MBR بوده و عموماً بانام MBR Magic Number و یا MBR Signature شناخته می شود.

```
|=====|
| First 440 bytes: Code Area |
|=====|
| 4 byte disk serial number | Used by Windows 2000
| 2 bytes null (0)         | and newer.
|=====|
| 16 byte partition table entry|
|=====|
| 16 byte partition table entry|
|=====|
| 16 byte partition table entry|
|=====|
| 16 byte partition table entry|
|=====|
| 2 byte MBR signature (0xAA55)|
|=====|
```

ساختر Master Boot Record

Byte Count	Description of contents

1	Boot indicator (0x00 for off, 0x80 for on)

3	Starting head, cylinder and sector

1	Filesystem descriptor

3	Ending head, cylinder and sector

4	Starting sector (relative to disk beginning)

4	Number of sectors in partition

عناصر تشکیل دهنده هر پارتیشن در MBR

Master Boot Record (offset)

0x0000 to 0x01BD - First 446 bytes
0x01BE to 0x01CD - Partition entry 1
0x01CE to 0x01DD - Partition entry 2
0x01DE to 0x01ED - Partition entry 3
0x01EE to 0x01FD - Partition entry 4
0x01FE to 0x01FF - Boot signature

آدرس اجزاء تشکیل دهنده MBR

۵- پارتیشن پایه (Primary Partition) :

حداکثر تعداد پارتیشن هایی که در یک سیستم مبتنی بر معماری Intel می توان درجدول پارتیشن موجود درسکتور راه انداز سیستم تعریف و ایجاد نمود، چهار پارتیشن می باشد که این پارتیشن ها را پارتیشن های پایه سیستم می نامیم و در لینوکس آنها را در اولین دیسک سخت سیستم به صورت hda1 تا hda4 می بینیم.

۶- پارتیشن منطقی (Logical Partition) :

یکی از روش های حل مشکل محدود بودن تعداد پارتیشن های موجود بر روی دیسک سخت، استفاده از یک پارتیشن توسعه یافته (Extended) است. در هر پارتیشن توسعه یافته امکان تعریف چندین پارتیشن منطقی وجود دارد که مشکل محدودیت تعداد پارتیشن های روی دیسک سخت را از این طریق می توان مرتفع کرد. ولی در هر صورت گنو/لینوکس محدودیتی را برای حداکثر تعداد پارتیشن هایی که بر روی یک دیسک سخت می توان ایجاد کرد، اعمال می کند. این مقدار در دیسک های SCSI حداکثر ۱۵ پارتیشن و در دیسک های IDE برابر ۶۳ پارتیشن در هر دیسک می باشد. نکته دیگری که باید در مورد پارتیشن های منطقی در نظر گرفت پیوستگی فضای مورد استفاده آن هاست که این امر در سایر پارتیشن های پایه سیستم امری الزامی نمی باشد. این مطلب نیز از اینجا متأثر می گردد که هر پارتیشن منطقی حاوی یک اشاره گر به پارتیشن منطقی بعدی می باشد. مشخصات هریک از این پارتیشن ها نیز در جدول پارتیشن توسعه یافته مربوط به خودشان که (Extended MBR) EMBR می باشد، قرار گرفته است. سیستم عامل های امروزی نیز در قبال پارتیشن های منطقی عملکردی مشابه با پارتیشن های پایه سیستم دارند.

Partition table	/dev/hda
Partition 1	/dev/hda1
Partition 2	/dev/hda2
Extended partition	/dev/hda3
Extended partition table	
Partition 3	/dev/hda5
Extended partition table	
Partition 4	/dev/hda6

ساختار پارتیشنهای اساسی و توسعه یافته دیسک

۷- پارتیشن فعال (Active Partition) :

همواره پارتیشنی که پرچم راه انداز آن (Bootable Flag) فعال باشد را به عنوان پارتیشن فعال دیسک می شناسیم، که تنها یک پارتیشن می تواند بر روی هاردیسک این خصوصیت را داشته باشد. پس ازاینکه BIOS محتویات MBR را به داخل حافظه بارگیری و اجرا نمود، MBR جدول پارتیشن موجود در خودش را چک کرده و پارتیشن فعال موجود در آن را به وسیله چک کردن پرچم راه انداز سایر پارتیشن ها پیدا می کند. سپس سکتور راه انداز پارتیشن فعال را که عموماً حاوی بارگذار راه اندازی چون GRUB یا NTLDR است را به داخل حافظه بارگیری کرده و اجرا می کند. برخی ازسیستم عامل ها مانند Windows 2000 الزاماً بایدبرروی پارتیشنی نصب گردندکه به عنوان پارتیشن فعال سیستم نیز عمل می کند.

قواعد نامگذاری دیسکهای سخت و پارتیشن ها

قواعدی که بارگذار راه انداز از آنها برای نامگذاری دیسکها و پارتیشن های آنها استفاده می کند ، متفاوت از نامگذاری سایر قطعات سخت افزاری است . بطور مثال شماره گذاری پارتیشن ها در اینجا از صفر شروع می شود . پس (hd0,0) اولین پارتیشن ، اولین دیسک سخت سیستم است . همچنین دیسک سختی که به عنوان Primary Master به سیستم متصل است نیز با نام /dev/hda شناخته می شود.

۴ پارتیشن فیزیکی را می توان با شماره های ۰ تا ۳ ایجاد کرد و در صورت ساخت پارتیشن های منطقی شماره آنها از ۴ شروع خواهد شد.

(hd0 , 0) : اولین پارتیشن فیزیکی از اولین دیسک سخت

(hd0 , 1) : دومین پارتیشن فیزیکی از اولین دیسک سخت

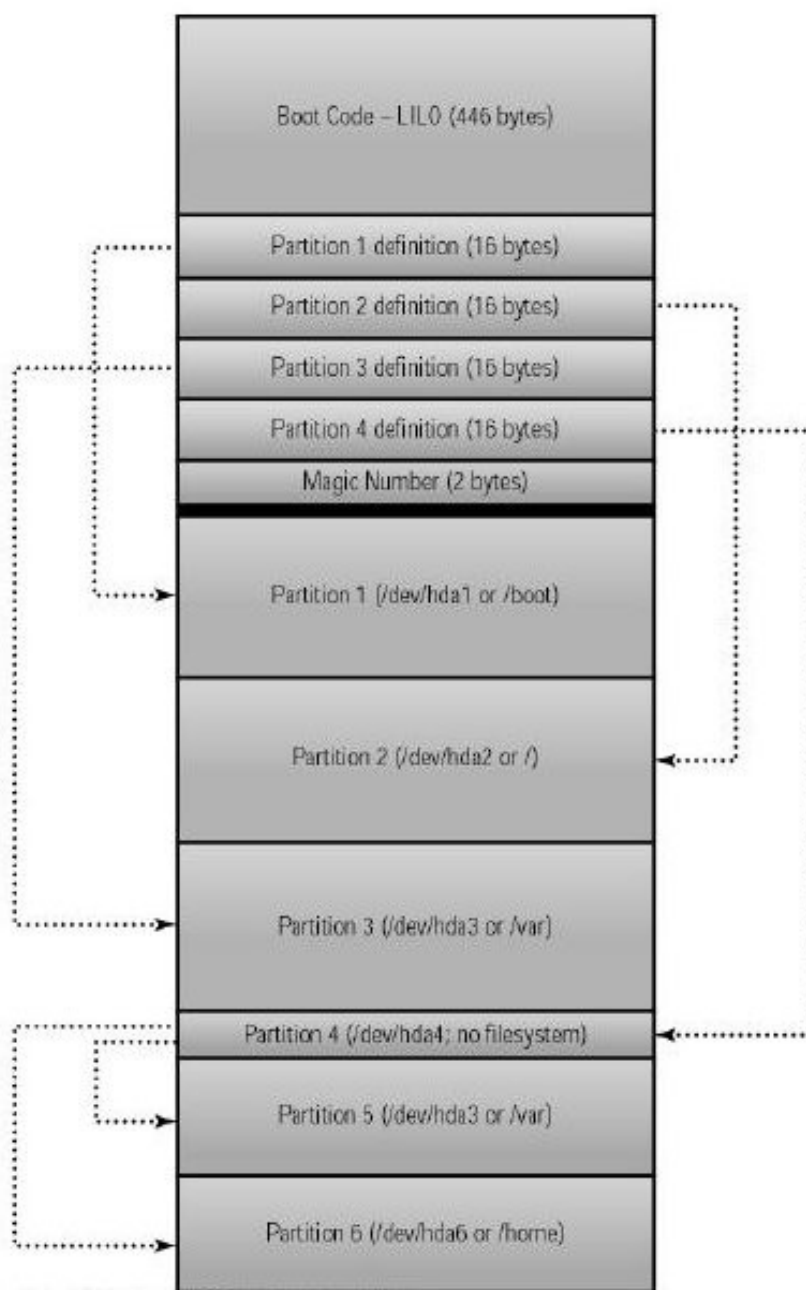
(hd0 , 2) : سومین پارتیشن فیزیکی از اولین دیسک سخت

(hd0 , 3) : چهارمین پارتیشن فیزیکی از اولین دیسک سخت (معمولا پارتیشن Extended می شود)

(hd0 , 4) : اولین پارتیشن منطقی اولین دیسک

(hd0 , 5) : دومین پارتیشن منطقی اولین دیسک

بارگذار راه انداز تفاوتی بین دیسکهای SCSI ، IDE و RAID قائل نمی شود . تمام دیسکهای سخت سیستم توسط BIOS یا دیگر کنترلرهای موجود ، براساس ترتیب بالا آمدن سیستم ، در BIOS شماره گذاری می گردند . متاسفانه بارگذار راه اندازها مانند GRUB قادر به نگاشت دقیق اسامی قطعات سخت افزاری سیستم به عنوان بخشی از ابزارهای BIOS نمی باشند . این کار در راه انداز GRUB توسط فایلی به نام device.map صورت می گیرد که در صورت لزوم می توان آنرا ویرایش کرد . مسیر کامل GRUB شامل نام قطعه ای از سیستم است که در داخل پرانتز نوشته شده است و مسیر دقیق فایل مربوطه را در سیستم فایل مشخص می کند .



ساختاری دیگر از دیسک سخت و آرایش پارتیشن‌ها

بخش دوم

تاریخچه گراب (GRUB:GRand Unified Bootloader)

GRUB نخستین بار در سال ۱۹۹۵ توسط اریک بولین ، در حالیکه سرگرم راه اندازی GNU Hurd در دانشگاه یوتا بود نوشته شد. بولین و برایان فورد مشخصه Multiboot بودن یک سیستم ، جهت برطرف کردن مشکل ناسازگاری بین روشهای مختلف راه اندازی یک PC را طراحی نمودند. سپس بولین شروع به اصلاح بارگذار راه انداز (Bootloader) برای FreeBSD نمود ، طوریکه آنرا تبدیل به یک بارگذار راه انداز Multiboot کرد . وی به سرعت متوجه شد که این کار بسیار راحتتر از نوشتن بارگذار راه انداز خودش ، که قصد داشت آنرا با استفاده از scratch و مبتنی بر بارگذار راه انداز FreeBSD تهیه کند ، است و بدین شکل اولین نسخه GRUB شکل گرفت. بولین امکانات فراوانی را به GRUB اضافه کرد ولی بدلیل اولویتهای کاری بالاتری که برای او وجود داشت نتوانست که آنرا به عنوان محصولی برای رفع نیاز کاربران توسعه دهد . در سال ۱۹۹۹ گوردن ماتزیگیت و یوشینوری کی اوکوجی GRUB را بعنوان یک بسته نرم افزار رسمی GNU ارائه دادند و توسعه آن با قرار دادن آخرین کدهای منبع موجود GRUB از طریق یک CVS (concurrent versioning system) غیررسمی صورت پذیرفت.

امکانات GRUB

GRUB قادر است سیستم فایل های موجود در یک سیستم را خوانده و تصاویری از هسته را که در سیستم وجود دارد ، در زمان راه اندازی سیستم تشخیص دهد . همچنین GRUB قابلیت پشتیبانی از انواع فایل سیستم ها از جمله سیستم فایل Reiser را نیز داراست . در ماشینهای دارای معماری x86 ، این بارگذار راه انداز ، محیطی با قابلیت و انعطاف پذیری بالا را قبل از راه اندازی سیستم عامل فراهم می کند که قابلیت جمع آوری اطلاعات متنوع در رابطه با سیستم را بر عهده دارد . این امر به کمک تهیه یک خط فرمان که امکانات فراوانی را در اختیار بارگذار راه انداز سیستم قرار می دهد فراهم می شود . از دیگر امکانات اصلی GRUB پشتیبانی از سیستم LBA می باشد . این سیستم در بسیاری از دیسکهای IDE و تمام دیسکهای SCSI مورد استفاده قرار می گیرد . تا قبل از توسعه LBA ، درایوهای دیسک سخت با محدودیت تعداد سیلندرهایی دیسک برای آدرس دهی مواجه بودند که حداکثر آن ۱۰۲۴ سیلندر بود .

این مقدار نیز برگرفته از روش آدرس دهی BIOS که ۱۰ بیتی است می باشد ($1024 = 2^10$) بدلیل وجود این مشکل BIOS قادر نبود تا فایلهایی را که در نقطه ای پس از آخرین سیلندر قابل آدرس دهی روی دیسک قرار دارند پیدا کند و در مواردی که این فایلها حاوی بارگذار راه انداز یا هسته یک سیستم عامل بودند ، امکان بارگذاری و استفاده از آن سیستم عامل عملاً وجود نداشت . سیستم LBA به GRUB و سایر بارگذار راه اندازها این امکان را می دهد که سیستم عامل های فراتر از محدوده سیلندر ۱۰۲۴ دیسک را بارگذاری

نمایند. فایل پیکربندی GRUB نیز هر بار در زمان راه اندازی سیستم از دیسک خوانده می شود که این امر از بازنویسی مجدد MBR پس از تغییر گزینه های راه اندازی جلوگیری می کند ، ولی در LILO آنطور که در فصول بعدی خواهید دید ، با تغییر یکی از پارامترهای فایل پیکربندی آن ، می بایست MBR سیستم را براساس پیکربندی جدید مجدداً بازنویسی نماییم که این کار نسبت به عملکرد GRUB در این زمینه بسیار مخاطره آمیز می باشد ، زیرا کوچکترین اشتباه در فایل پیکربندی موجب می گردد تا MBR به شکلی غلط پیکربندی شود و سیستم قابلیت راه اندازی خود را از دست بدهد . ولی در GRUB براحتی می توانیم با استفاده از خط فرمان و دستورات آن پیکربندی را اصلاح نماییم . همچنین پس از اعمال تغییرات به فایل پیکربندی GRUB ، نیازی به راه اندازی مجدد GRUB نبوده و کلیه تغییرات به شکل خودکار توسط GRUB تشخیص داده می شوند .

از دیگر امکانات اصلی GRUB می توان به موارد زیر اشاره کرد :

۱ - استفاده همزمان چندین سیستم عامل در نسخه های گوناگون بر روی یک یا چند دیسک سخت

۲ - امکان راه اندازی سیستم عامل تحت شبکه بوسیله پروتکل های DHCP و BOOTP جهت پشتیبانی از روشهای Multi boot تحت شبکه

۳ - دسترسی به ساختار دیسک (Disk Geometry) ، خواندن حافظه و شناسایی پردازنده و پورتهای I/O

۴ - امکان استفاده از کلمه عبور بر روی بارگذار راه انداز و پشتیبانی از فایل های فشرده

سه فایل اصلی GRUB نیز که پیکربندی آن براساس آنها صورت می گیرد عبارتند از:

(۱) `/boot/grub/menu.lst` : حاوی اطلاعات مربوط به پارتیشن ها یا سیستم عامل هایی است که سیستم می تواند بوسیله آنها راه اندازی

شود.

(۲) `/etc/grub.conf` : این فایل حاوی پارامترهایی است که پوسته GRUB جهت نصب صحیح بارگذار راه انداز به آنها نیاز دارد .

(۳) `/boot/grub/device.map` : فایل مذکور نام قطعاتی که در GRUB و BIOS ذکر شده اند را به نامهای معادلشان در گنو/لینوکس تبدیل

می کند. در واقع مترجمی بین سیستم عامل و BIOS می باشد که توسط آن دیگر فایل های پیکربندی ، پارتیشن های سیستم را تشخیص می دهند.

نحوه تهیه و نصب GRUB

قبل از نصب GRUB می بایست از وجود آخرین نسخه آن آگاهی پیدا کرده و بسته مربوط به آنرا ترجیحاً در قالب rpm دانلود نماییم .

جهت تهیه این نرم افزار می توانید از سایتهایی چون rpmfind.net و یا rpm.pbone.net استفاده نمایید. پس از دانلود بسته تحت کاربر

ریشه و توسط دستور **rpm** در کنسول ، GRUB را نصب کرده و فرمان زیر را در SHELL اجرا می کنیم :

```
# /sbin/grub-install <location>
```

location آدرس دیسکی است که قرار است GRUB در آن نصب شود . به طور مثال جهت نصب GRUB در (Master Boot) MBR

Record اولین دیسک سخت سیستم از دستور زیر استفاده می کنیم :

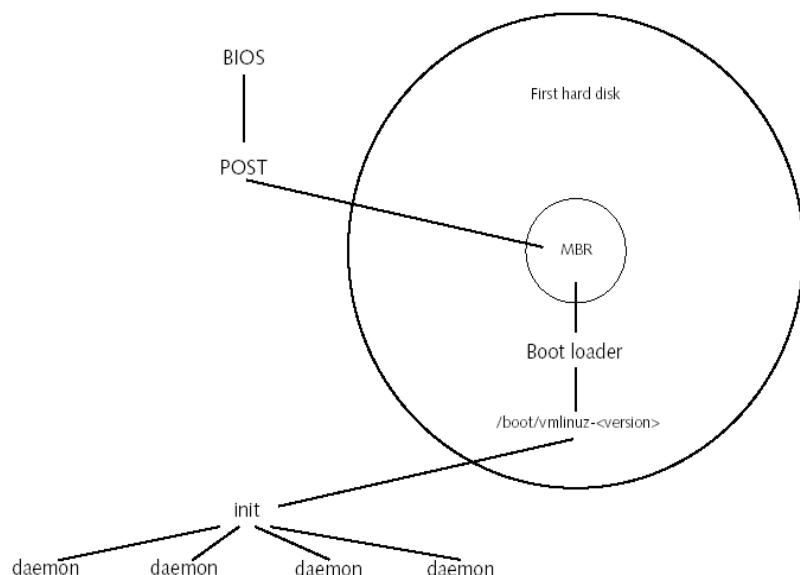
```
# /sbin/grub-install /dev/hda
```

دنبال کردن رویه راه اندازی سیستم

زمانیکه سیستم خود را روشن می کنید ، BIOS شروع به تست کردن سایر قطعات سخت افزاری موجود جهت راه اندازی سیستم می کند که همانطور در مطالب گذشته عنوان شد نام این مرحله POST می باشد . سپس BIOS یکی از اجزا قابل راه اندازی سیستم شامل فلاپی ، دیسک سخت ، حافظه های Flash و یا ابزارهای شبکه را جهت راه اندازی سیستم از روی آن انتخاب می کند که این وسیله راه انداز معمولاً در SETUP سیستم به عنوان گزینه پیش فرض در کنار ترتیب راه اندازی سایر اجزا ، مشخص گردیده است . BIOS اولین سکتور وسیله راه انداز که سکتور راه انداز (Boot Sector) نامیده می شود را خوانده و بدین طریق اطلاعات کاملی از ساختار پارتیشن های سیستم و مشخصات آنها را کسب می کند . همانطور که گفته شد سکتور راه انداز سیستم حاوی برنامه کوچکی است که وظیفه خواندن سیستم عامل از روی دیسک و راه اندازی آنرا برعهده دارد . زمانیکه جهت راه اندازی سیستم عاملی مثل گنو/لینوکس از یک فلاپی دیسک استفاده می کنیم ، سکتور راه انداز حاوی کدی است که تنها چند بلوک ابتدایی دیسک که به محل مشخصی از حافظه اشاره می کنند را می خواند . از آنجا که در یک فلاپی راه انداز گنو/لینوکس سیستم فایلی وجود ندارد ، هسته سیستم عامل در سکتورهای متوالی ذخیره می شوند که این امر رویه راه اندازی سیستم را تسهیل می بخشد . البته می توان در مواردی هم از فلاپی های حاوی یک سیستم فایل که شامل راه اندازهای LILO و یا GRUB هستند استفاده کرد . هنگامیکه سیستم از یک دیسک سخت راه اندازی می گردد ، کد موجود در سکتور راه انداز اصلی دیسک (MBR) جدول پارتیشن دیسک را بررسی کرده و پارتیشن فعال دیسک را تشخیص می دهد ، سپس سکتور راه انداز آن پارتیشن را خوانده و برنامه موجود در سکتور راه انداز آنرا اجرا می کند .

عملیاتی که توسط کد موجود در این سکتور راه انداز انجام می گیرد دقیقاً همان کاری است که توسط سکتور راه انداز فلاپی دیسک انجام شدنی است . همچنین از آنجا که تخصیص یک پارتیشن مجزا برای هسته سیستم عامل کار چندان جالبی به نظر نمی رسد و از طرفی هم برنامه موجود در سکتور راه انداز سیستم نمی تواند به صورت ترتیبی شروع به خواندن کل فضای دیسک جهت پیدا کردن و بارگیری هسته

سیستم عامل نماید ، وجود ابزاری به نام بارگذار راه انداز (Boot Loader) در اینجا خود را الزامی می نماید .



رویه بارگیری بارگذار راه انداز توسط BIOS

بارگذار راه انداز سیستم به شکل پیش فرض سراغ هسته از پیش تعیین شده سیستم رفته و آنرا بارگیری می کند که در این حالت امکان انتخاب یکی از چندین هسته موجود جهت بارگیری به داخل حافظه وجود دارد .

پس از اینکه هسته یک توزیع گنو/لینوکس خوانده شده و به داخل حافظه بارگیری گردید ، مراحل زیر صورت می گیرد :

۱- از آنجا که هسته گنو/لینوکس به شکل فشرده نصب می گردد ، در ابتدا این هسته توسط برنامه کوچکی که در ابتدای فایل مربوط به آن قرار دارد، خودش را از حالت فشرده خارج می کند .

۲- سپس هسته کارت گرافیکی سیستم را شناسایی کرده و مد کاری آنرا مشخص می کند . تعیین مد کاری نیز می تواند با سؤال از کاربر انجام پذیرد . انجام این عملیات در زمان کامپایل هسته و یا با استفاده از سرویس udev نیز امکان پذیر است .

۳- در مرحله بعد هسته سایر قطعات سخت افزاری موجود شامل کارت شبکه، فلاپی درایو، کارت صدا و دیگر اجزا را چک کرده و پیکربندی آنها را از طریق بارگیری درایور هر یک از این قطعات در حافظه انجام می دهد .

معمولاً اطلاعات مربوط به این مرحله را می توان با مراجعه به فایل messages موجود در مسیر /var/log و یا با استفاده از دستور dmesg

\$ dmesg | less

به شکل زیر مشاهده کرد :

حال با مشاهده خروجی فرمان فوق می توانیم دقیقاً عملکرد هسته را در زمان راه اندازی سیستم مشاهده کنیم .

```
Linux version 2.6.12.2 (root@linux) (gcc version 3.3.5 20050117 (prerelease) (SUSE Linux))
#19 SMP Mon Nov 14 06:55:01 MST 2005
BIOS-provided physical RAM map:
BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
BIOS-e820: 0000000000100000 - 000000001fff0000 (usable)
BIOS-e820: 000000001fff0000 - 000000001fff3000 (ACPI NVS)
BIOS-e820: 000000001fff3000 - 0000000020000000 (ACPI data)
BIOS-e820: 00000000fec00000 - 0000000100000000 (reserved)
0MB HIGHMEM available.
511MB LOWMEM available.
found SMP MP-table at 000f5210
On node 0 totalpages: 131056
  DMA zone: 4096 pages, LIFO batch:1
  Normal zone: 126960 pages, LIFO batch:31
  HighMem zone: 0 pages, LIFO batch:1
DMI 2.3 present.
ACPI: RSDP (v000 GBT                          ) @ 0x000f6be0
ACPI: RSDT (v001 GBT      AWRDACP1 0x42302e31 AWRD 0x01010101) @ 0x1fff3000
ACPI: FADT (v001 GBT      AWRDACP1 0x42302e31 AWRD 0x01010101) @ 0x1fff3040
ACPI: MADT (v001 GBT      AWRDACP1 0x42302e31 AWRD 0x01010101) @ 0x1fff7000
ACPI: DSDT (v001 GBT      AWRDACP1 0x00001000 MSFT 0x0100000c) @ 0x00000000
ACPI: PM-Timer IO Port: 0x1008
ACPI: Local APIC address 0xfee00000
lines 1-24
```

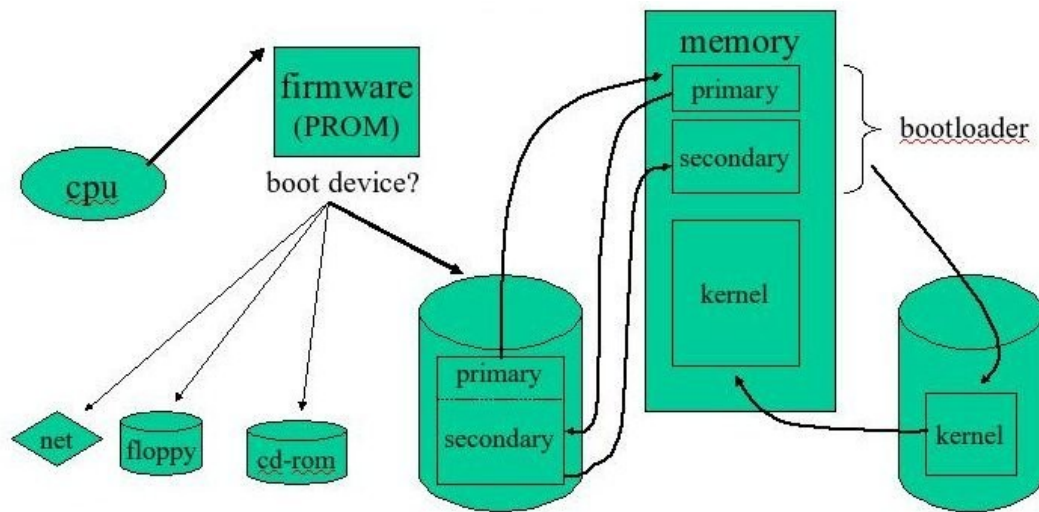
صفحه اول خروجی فرمان \$ dmesg | less

۴- هسته پس از شناسایی سایر قطعات سخت افزاری سیستم، اقدام به اتصال سیستم فایل ریشه سیستم می کند، که محل آن توسط بارگذار راه انداز سیستم تعیین می شود. نوع فایل سیستم نیز به شکل خودکار توسط هسته تشخیص داده می شود. اگر عملیات اتصال فایل سیستم ریشه به هر دلیلی صورت نگیرد، معمولاً سیستم مجدداً راه اندازی شده و یا پیغام kernel panic نمایش داده می شود. فایل سیستم ریشه نیز عموماً به صورت فقط خواندنی به سیستم اتصال می یابد که در زمان راه اندازی سیستم عملیات چک کردن آن نیز توسط ابزاری چون fsck نیز صورت می گیرد. (اتصال فایل سیستم ریشه به هیچ وجه به شکل خواندنی-نوشتنی در زمان راه اندازی سیستم کار صحیحی نمی باشد)

۵- پس از اتمام مراحل مذکور، هسته اقدام به اجرای برنامه init که در مسیر /sbin/init واقع شده است می کند. فرآیند init عملیات مربوط به راه اندازی سایر سرویسها و برنامه های پیش فرض سیستم را انجام می دهد. تمام این برنامه ها به صورت فرآیندهای

پشت صحنه (Background) انجام می شوند .

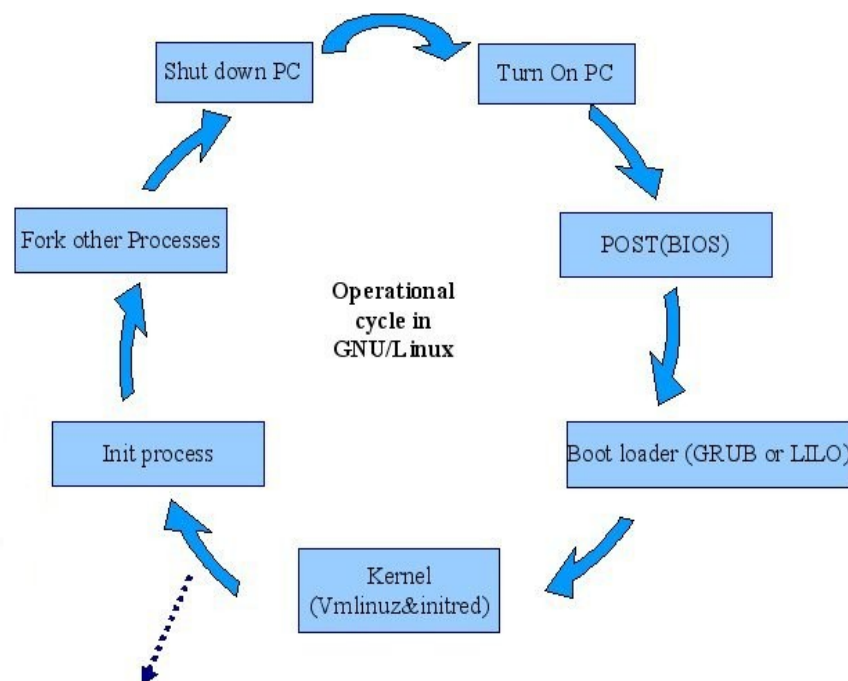
۶- سپس فرآیند init به مد چند کاربره (Multi-user) سوچ کرده و یک کنسول مجازی را از طریق برنامه getty ایجاد می کند . این برنامه به کاربران اجازه login از طریق کنسولهای مجازی و ترمینالهای سریال را می دهد . در این زمان فرآیند init ممکن است که اقدام به اجرای برنامه های دیگری مانند یک Display manager و یا اجرای یک محیط گرافیکی X نماید .



ساختار بلوکی

بارگیری هسته به داخل حافظه

با اتمام مراحل فوق، فرآیند راه اندازی سیستم با موفقیت صورت گرفته و سیستم در وضعیت پایداری قرار خواهد گرفت.



Linuxrc (در درخت initrd اجرا شده و کارش معرفی device ها و فایل سیستم به kernel است)

چرخه عملیاتی در گنو/لینوکس

GRUB به عنوان بارگذار راه انداز سیستم

GRUB متشکل از دو بخش اساسی است. یک بخش آن که stage ۱ نام دارد شامل ۵۱۲ بایت بوده و در MBR پارتیشن دیسک سخت یا فلاپی دیسک قرار میگیرد. این قسمت وظیفه بارگیری بخش دوم GRUB، که stage 2 است را برعهده دارد که عملیات اصلی بارگذار راه انداز را انجام می دهد. در بین این دو مرحله، مرحله ای با نام stage 1.5 نیز وجود دارد که در مسیر boot/grub / قرار داشته و وظیفه بارگیری file system ها را در زمان راه اندازی بر عهده دارد. سیستم فایل‌هایی چون Minix، Reiserfs، Ext3، Ext2 و فایل سیستم های ویندوز مثل FAT، بوسیله Stage2 قابل پشتیبانی هستند؛ همچنین از سیستم فایل‌های خاصی چون UFS، XFS، JFS و FFS که توسط سیستم های BSD استفاده می شوند نیز می توان جهت دسترسی و مدیریت فایل ها و منابع موجود در دیسک سخت بهره برداری کرد. از GRUB نسخه ۰.۹۵ به بعد قابلیت بالا آمدن سیستم عامل از CD یا DVD حاوی یک سیستم فایل استاندارد ISO9660 نیز میسر گردید و از تمام اجزائی که این قابلیت را دارا بوده و توسط BIOS تشخیص داده می شوند، می توان به عنوان یک ابزار Boot سیستم استفاده نمود.

جهت اعمال هر تغییر در فایل پیکربندی GRUB، نیازی به نصب مجدد آن نمی باشد، بلکه در هر بار بالا آمدن سیستم، GRUB فایل پیکربندی خود را که menu.lst است، خوانده و پارتیشن های حاوی اطلاعات مربوط به هسته سیستم عامل را بر اساس مسیرهای موجود در آن بازیابی می کند. GRUB را می توان به روشهای مختلفی کنترل کرد. گزینه های موجود در فایل پیکربندی grub را در زمان راه اندازی سیستم می توان از منوی گرافیکی (Splash screen) انتخاب کرد که از فایل menu.lst خوانده می شوند. همچنین از طریق منوی Boot می توان به صورت محاوره ای فرامینی را جهت راه اندازی سیستم عامل وارد کرد. در واقع از این طریق می توان حتی سیستم عامل های نصب شده ای را که گزینه ای برای راه اندازیشان در فایل menu.lst تعریف نشده است را بارگذاری کند.

بررسی ساختار فایل menu.lst

کلیه گزینه های موجود در منوی راه اندازی سیستم مبتنی بر داده های موجود در فایل `/boot/grub/menu.lst` می باشد که شامل اطلاعاتی درباره تمام پارتیشن ها یا سیستم عامل هایی است که می توان آنها را در زمان راه اندازی سیستم انتخاب کرد. در هر بار راه اندازی سیستم، GRUB فایل مذکور را از دیسک سخت بارگذاری می کند. فایل `menu.lst` حاوی فرامینی است که شکل آنها بسیار ساده است. هر خط این فایل شامل یک فرمان به همراه مجموعه ای از پارامترهای اختیاری است که همانند دستورات Shell با `blank` از هم جدا می شوند. توضیحات در فایل با کاراکتر `#` مشخص می شوند. جهت تشخیص گزینه های موجود در فایل `menu.lst`، برای هر قسمت یک `title` در نظر گرفته شده است. متنی که پس از عبارت `title` قرار می گیرد به عنوان یکی از گزینه های قابل انتخاب در منوی Boot سیستم خواهد بود.

کلیه فرامینی که پس از عبارت `title` قرار می گیرند، با انتخاب گزینه موجود در منوی Boot به ترتیب اجرا خواهند شد. ساده ترین شکل آن بارگذاری و استفاده از بارگذار راه انداز دیگر سیستم عامل های موجود سیستم است. فرمان `chainloader` این کار را با اشاره به بلوک راه انداز دیگر پارتیشن های سیستم امکانپذیر می سازد، مانند: `chainloader (hd0,3)+1`

از جمله فرامین موجود در این بخش فرمان `kernel` است که جهت تعیین محل تصویر هسته (`kernel image`) گنو/لینوکس استفاده می شود. اولین پارامتر این فرمان، مسیر `image` در یکی از پارتیشنهای دیسک سخت سیستم است. همچنین با پارامتری به نام `initrd` در فایل `menu.lst` برخورد می کنیم که جزئی از هسته سیستم می باشد. در هسته بخش کوچکی شامل یک فایل سیستم وجود دارد که قبل از اتصال فایل سیستم ریشه، این بخش به داخل حافظه RAM رفته و در آنجا اجرا میگردد. این جزء که `initrd` (initial ram disk) نام دارد، در مراحل اولیه boot سیستم، توسط هسته بعنوان یک فایل سیستم ریشه موقتی مورد استفاده قرار می گیرد و محتویات آن شامل یک `Minimal Linux System` است که کنترل سیستم را به شکل موقت برعهده می گیرد. اگر هسته فاقد درایورهای لازم جهت دسترسی به پارتیشن ریشه سیستم باشد، `initrd` را می بایست توسط دستور GRUB مشخص کرد، که تنها پارامتر آن مسیر فایل `initrd` است. همچنین از آنجا که آدرس محل بارگذاری `initrd` نیز در تصویر هسته بارگذاری شده وجود دارد، فرمان `initrd` را نیز باید بلافاصله پس از فرمان `kernel` قرار دهیم. توسط فرمان `root` می توان مشخصات فایل های هسته و `initrd` را جهت قرار دادن در `menu.lst` خلاصه کرد. تنها پارامتر فرمان `root`، وسیله یا پارتیشنی است که GRUB روی آن نصب شده است. آدرس وسیله ای که پس از عبارت `root` قرار میگیرد، برای هسته، `initrd` و تمام قطعاتی که مسیر آنها بطور دقیق مشخص نشده است، تا کلمه `root` بعدی در فایل `menu.lst` بکار می رود. دستور boot در انتهای هر کدام از اجزای منوی راه اندازی سیستم به شکل خودکار اجرا می گردد، پس نیازی به قرار دادن آن در

فایل menu.lst نمی باشد. در صورتیکه جهت راه اندازی سیستم ، از خط فرمان GRUB به صورت محاوره ای استفاده کنیم ، در انتهای خط فرمان می بایست فرمان boot را وارد نماییم. این فرمان به تنهایی فاقد هر گونه پارامتری بوده و به ندرت از آن برای بارگذاری تصویر هسته و یا یک chainloader خاص استفاده می شود .

یک مثال از فایل menu.lst

مثال زیر ساختاری از فایل menu.lst را نمایش می دهد که در آن یک پارتیشن گنو/ لینوکسی در /dev/hda5 ، یک پارتیشن ریشه در /dev/hda7 و یک پارتیشن ویندوزی در /dev/hda1 پیکربندی شده است .

```
gfxmenu (hd0,4)/message
```

```
color white/blue black/light-gray
```

```
default 0
```

```
timeout 8
```

```
title linux
```

```
kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
```

```
initrd (hd0,4)/initrd
```

```
title windows
```

```
chainloader(hd0,0)+1
```

```
title floppy
```

```
chainloader(fd0)+1
```

```
title failsafe
```

```
kernel (hd0,4)/vmlinuz.shipped root=/dev/hda7 ide=nodma apm=off acpi=off vga=normal nosmp maxcpus=0 3
```

```
initrd (hd0,4)/initrd.shipped
```

شرح فایل فوق:

اولین بلوک ، پیکربندی صفحه پس زمینه زمان راه اندازی سیستم (splash screen) را مشخص می کند :

gfxmenu (hd0,4)/message

متن مربوط به عکس پس زمینه در مسیر /dev/hda5 قرار گرفته است .

color white/blue black/light-gray

سطر فوق نیز الگوی رنگ صفحه ی راه اندازی را مشخص می کند :

رنگ متن (سفید) ، رنگ زمینه (آبی) ، نوار انتخاب (مشکی) و پس زمینه قسمت انتخاب شده (خاکستری روشن) . برخی دیگر از

رنگهایی که می توان از آنها در این بخش استفاده کرد عبارتند از :

dark-gray , light-gray , light-blue , light-red, light-green , light-cyan , light-magenta , yellow

default 0

بیانگر این است که اولین گزینه منوی راه اندازی ، به عنوان پیش فرض انتخاب شده است .

timeout 8

پس از ۸ ثانیه ، اگر کاربر گزینه ای را انتخاب نکرد، سیستم به صورت خودکار از روی گزینه پیش فرض راه اندازی می شود.

دومین و بزرگترین بلوک ، لیستی از سیستم عامل های قابل راه اندازی توسط سیستم را مشخص می کند . بخش مربوط به هر سیستم

عامل بوسیله یک title انتخاب می شود :

(۱) اولین گزینه (title linux) مربوط به راه اندازی توزیع گنو/ لینوکسی موجود می باشد . در این مثال هسته (فایل vmlinuz) در اولین

پارتیشن منطقی اولین دیسک سخت سیستم (پارتیشن راه اندازی) قرار گرفته است . پارامترهای هسته مانند پارتیشن ریشه و مد کارت

گرافیکی سیستم نیز در اینجا قرار داده شده اند . محل پارتیشن ریشه نیز براساس قواعد نامگذاری گنو/لینوکس مشخص شده است

(dev/hda7) . از آنجا که این اطلاعات توسط هسته سیستم عامل گرفته می شوند و با GRUB ارتباطی ندارند ، از این قواعد نامگذاری

استفاده می شود . فایل initrd ، از دیگر فایل های مورد نیاز هسته نیز در همین پارتیشن قرار دارد .

(۲) دومین گزینه جهت بارگیری ویندوز بکار می رود . ویندوز نیز از اولین پارتیشن اولین دیسک سخت سیستم (0 ، hd0) راه اندازی

میشود . دستور chainloader + 1 نیز موجب می گردد تا GRUB اولین سکتور پارتیشن مشخص شده را خوانده و اجرا کند .

(۳) گزینه بعدی راه اندازی سیستم از طریق دیسک نرم بدون اعمال تغییرات در تنظیمات BIOS سیستم را میسر می سازد .

۴) انتخاب گزینه failsafe ، امکان راه اندازی گنو/لینوکس توسط انتخاب پارامترهای خاصی از هسته ، در زمان راه اندازی سیستم ، در صورت وجود مشکل در راه اندازی را در اختیار قرار می دهد .

فایل menu.lst را هر زمان که نیاز باشد می توان اصلاح کرد . پس از اصلاح GRUB تنظیمات جدید آن در راه اندازی بعدی سیستم اعمال خواهند شد .

نمونه دیگری از یک فایل menu.lst

ویرایش اجزاء منوی راه اندازی در طول رویه بالا آمدن سیستم

در منوی GRUB در زمان راه اندازی سیستم ، با استفاده از کلیدهای مکان نما می توان سیستم عامل مورد نظر برای راه اندازی را انتخاب کرد . در این زمان کاربران سیستم قادرند در اعلان مربوط به هر گزینه فرامینی را اجرا کرده و پارامترهایی را در گزینه ها تغییر دهند . جهت ویرایش مستقیم گزینه ها ابتدا جهت خروج از صفحه splash کلید Esc را فشار داده و سپس حرف E فشار می دهیم . تغییراتی را

که در این هنگام به رویه راه اندازی سیستم اعمال می کنیم تنها در پروسه راه اندازی جاری مورد استفاده قرار می گیرند و به شکل پایدار در زمان راه اندازی سیستم مورد نظر مورد استفاده قرار نمی گیرند .



صفحه آغازین بارگذار راه انداز GRUB پس از روشن شدن سیستم

نکته : تنها طرحی از صفحه کلید (keyboard layout) که در زمان راه اندازی سیستم ، قابل استفاده می باشد ، طرح US است .

پس از ورود به مد ویرایش باید گزینه مورد نظر جهت ویرایش را انتخاب کرده و سپس مسیر یا پارتیشنی را که در رویه راه اندازی مشکل

ایجاد می کند، پیدا کرده و آنرا اصلاح نماییم. پس از اصلاح گزینه مورد نظر با فشردن کلید Enter از مد ویرایش خارج شده و به منوی اصلی برمی گردیم. با فشردن کلید B این گزینه جهت راه اندازی سیستم اعمال می شود .

جهت اعمال همیشگی تغییرات مورد نظر در زمان راه اندازی ، فایل menu.lst را بعنوان کاربر ریشه باز کرده و پارامترهای مورد نظرمان برای سیستم عامل ها و هسته های موجود را وارد نموده و پس از ذخیره سازی تغییرات از آن خارج می شویم :

```
title linux
```

```
kernel (hd0,0)/vmlinuz root=/dev/hda3 (additional parameter)
```

```
initrd (hd0,0)/initrd
```

GRUB به شکل خودکار پارامترهای جدید را در زمان راه اندازی بعدی سیستم در نظر می گیرد . تنها نکته ای که در زمان تنظیم پارامترها باید در نظر بگیریم ، جدا کردن آنها با «فاصله» از یکدیگر است .

استفاده از Wild Card ها جهت انتخاب هسته در زمان راه اندازی

در مواردی برای استفاده از هسته های گنو/ لینوکسی که به شکل سفارشی ساخته شده اند ، جهت استفاده از هسته جدید و فایل های initrd، نیاز به تغییر اجزاء فایل menu.lst یا ویرایش خط فرمان در زمان راه اندازی داریم . برای تسهیل این کار ، می توانیم از wild card ها جهت به روز رسانی خودکار لیست هسته های موجود در GRUB استفاده نماییم . تصاویری (kernel image) از هسته که با الگوی خاصی مطابقت دارند ، به شکل خودکار به لیست تصاویر قابل راه اندازی هسته افزوده می گردند . فعالسازی wild card ها را می توان با استفاده از پارامترهایی در فایل menu.lst انجام داد . استفاده از نام و شناسه یکسان برای تمام تصاویر هسته و فایل های initrd مربوط به آنها ، جهت تطبیق هر هسته با initrd متناظرش در پروسه راه اندازی سیستم ، کار بسیار مفیدی می باشد . مثال زیر این مطلب را نشان می دهد :

```
initrd-default
```

```
initrd-test
```

```
vmlinuz-default
```

```
vmlinuz-test
```

در این حالت شما می توانید دو تصویر از هسته را در فایل پیکربندی GRUB قرار دهید . جهت استفاده از گزینه های linux- و linux-test default در حین راه اندازی ، نیاز به افزودن اجزاء زیر به فایل menu.lst داریم :

```
title linux-*
```

```
wildcard (hd0,4)/vmlinuz-*
```

```
kernel (hd0,4)/vmlinuz-* root=/dev/hda7 vga=791
```

```
*-initrd (hd0,4)/initrd
```

در این مثال ، GRUB پارتیشن (hd0,4) را برای پیدا کردن گزینه های مورد نظرش بوسیله wild card جستجو می کند . این گزینه ها در ایجاد گزینه های جدید منوی GRUB بکار گرفته می شوند . در مثال قبل ، GRUB در صورت وجود عبارات مذکور در فایل menu.lst ، به صورت زیر عمل میکند :

```
title linux-default
```

```
wildcard (hd0,4)/vmlinuz-default
```

```
kernel (hd0,4)/vmlinuz-default root=/dev/hda7 vga=791
```

```
initrd (hd0,4)/initrd-default
```

```
title linux-test
```

```
wildcard (hd0,4)/vmlinuz-test
```

```
kernel (hd0,4)/vmlinuz-test root=/dev/hda7 vga=791
```

```
initrd (hd0,4)/initrd-test
```

فایل device.map

همانطور که قبلا اشاره شد فایل device.map اسامی اجزاء سخت افزاری موجود در GRUB را به اسامی قطعات با فرمت گنو/لینوکس تبدیل می کند . در سیستم هایی که شامل دیسکهای IDE و SCSI هستند ، GRUB رویه راه اندازی سیستم را به نحو خاصی باید تعیین کند ، از آنجایی که در زمان راه اندازی سیستم BIOS به اطلاعات موجود در GRUB هیچ گونه دسترسی ندارد ، جهت انجام پروسه راه

اندازی سراغ فایل `/boot/grub/device.map` می رود . بطور مثال برای سیستمی که در پروسه راه اندازی آن دیسک IDE قبل از SCSI قرار می گیرد ، فایل `device.map` را به صورت زیر می توان نمایش داد :

```
(fd0) /dev/fd0
```

```
(hd0) /dev/hda
```

```
hd1) /dev/sda)
```

ترتیب دیسکهای SCSI ، IDE و سایر دیسکهای سیستم به پارامترهای مختلفی بستگی دارد و گنو/ لینوکس قادر به تبدیل اسامی دیسکها از فرمت BIOS به فرمت خودش نیست. اجزاء موجود در فایل `device.map` توسط کاربر ریشه نیز قابل تنظیم است . اگر در زمان راه اندازی سیستم با مشکلی مواجه شدیم ، رویه راه اندازی سیستم را از داخل این فایل می توان چک کرده و یا با استفاده از GRUB Shell ، آنطور که در بخشهای بعدی توضیح داده خواهد شد ، در زمان راه اندازی می توان به اصلاح موقتی سایر اجزاء راه اندازی سیستم پرداخت. پس از راه اندازی سیستم تغییرات را به شکل پایدار در فایلهای پیکربندی سیستم می توان اعمال نمود . با انجام تنظیمات جدید در فایل `device.map` ، بارگذار راه انداز GRUB را می بایست مجدداً نصب نماییم . این امر موجب بارگیری مجدد فایل `device.map` و اجرای فرامین موجود در فایل `grub.conf` می گردد :

```
grub --batch < /etc/grub.conf
```

فایل `/etc/grub.conf`

سومین فایل مهم پیکربندی GRUB که برگرفته از فایل های `menu.lst` و `device.map` است ، فایل `grub.conf` می باشد . این فایل حاوی پارامترها و گزینه هایی است که GRUB جهت نصب کامل بارگذار راه انداز سیستم به آنها نیاز دارد :

```
root (hd0,4)
```

```
install /grub/stage1 d (hd0) /grub/stage2 0x8000 (hd0,4)/grub/menu.lst
```

```
quit
```

مفهوم هر یک از پارامترهای فوق عبارتند از :

۱- `root (hd0,4)`: فرامینی که پس از این عبارت قرار می گیرند ، مکان فایل های لازم جهت راه اندازی سیستم را مشخص می کنند که در

این مثال اولین پارتیشن منطقی از اولین دیسک سخت سیستم است .

۲- **install parameter** : دستور grub با پارامتر `install.stage1` ، جهت نصب بارگذار راه انداز در MBR اولین دیسک سخت سیستم بکار می رود (`(hd0) d /grub/stage1`). مرحله دوم یا stage2 نیز باید بنا به دستور فوق الذکر در آدرس 0x8000 حافظه بارگیری شود . آخرین جزء فرمان install محل قرار گرفتن فایل menu.lst جهت استفاده GRUB را مشخص می کند .

پوسته GRUB

نرم افزار GRUB عموماً به ۲ شکل مورد استفاده قرار می گیرد :

ابتدا به عنوان یک بارگذار راه انداز و سپس به عنوان یک برنامه گنو/ لینوکسی در مسیر `/usr/sbin/grub` . این برنامه به عنوان پوسته ای برای GRUB مورد استفاده قرار می گیرد . توانایی نصب GRUB به عنوان یک بارگذار راه انداز بر روی دیسک سخت یا فلاپی به شکلی فشرده و کارآمد در فرامین `install` و `setup` گردآوری شده است . این دستورات در زمان بارگیری سیستم عامل در پوسته GRUB قابل دسترسی هستند . در هر صورت فرامین `install` و `setup` قبل از راه اندازی گنو/ لینوکس در طول رویه راه اندازی سیستم در دسترس سایر کاربران سیستم می باشند . این امر بازسازی سیستم های آسیب دیده ای را که به هیچ وجه قابل راه اندازی نیستند ، تسهیل می بخشد زیرا از این راه می توان فایل های پیکربندی بارگذار راه انداز سیستم را با دادن پارامترهای صحیح اصلاح نمود . اعمال تنظیمات دستی در طول رویه راه اندازی سیستم نیز برای آزمایش پارامترهای جدید، بدون اعمال آنها به سیستم اصلی ، امری مفید می باشد . بطور مثال برای آزمایش یک هسته جدید ، می توان فرمان `kernel` را به همراه مسیر هسته جدید وارد کرد . به همین شکل می توان سیستم های دچار مشکل را علی رقم وجود مشکل در فایل `menu.lst` آنها ، مجدداً راه اندازی نمود . تخصیص اسامی قطعات موجود در GRUB به اسامی اجزاء سخت افزاری گنو/لینوکس نیز تنها با استفاده از پوسته GRUB صورت می گیرد که برای انجام آن از فایل `device.map` استفاده میکند .

تنظیم یک گذرواژه برای بارگذار راه انداز

عموماً دسترسی به یک سیستم فایل و منابع آن ، توسط GRUB و قبل از راه اندازی سیستم عامل امکانپذیر است . کاربرانی که فاقد مجوزهای دسترسی لازم هستند قادر خواهند بود بدینوسیله به فایل ها و منابعی که جواز دسترسی به آنها را دارا نیستند ، دست یابند . جهت جلوگیری از این نوع دسترسی غیرمجاز توسط کاربران سیستم در زمان بالا آمدن یک سیستم عامل خاص ، از `boot password`

استفاده می کنیم. جهت قراردادن گذرواژه بر روی GRUB به عنوان کاربر ریشه سیستم مراحل زیر را دنبال می کنیم :

(۱) فرمان grub را تحت کاربر ریشه اجرا می کنیم .

(۲) توسط فرمان md5crypt گذرواژه رمزگذاری شده موردنظرمان را بدست می آوریم.

```
#grub
```

```
grub > md5crypt
```

```
password : *****
```

```
Encrypted : $1$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```

(۳) رشته رمزگذاری شده ای که در مرحله قبل بدست آمده است را در بخش تنظیمات عمومی menu.lst قرار می دهیم:

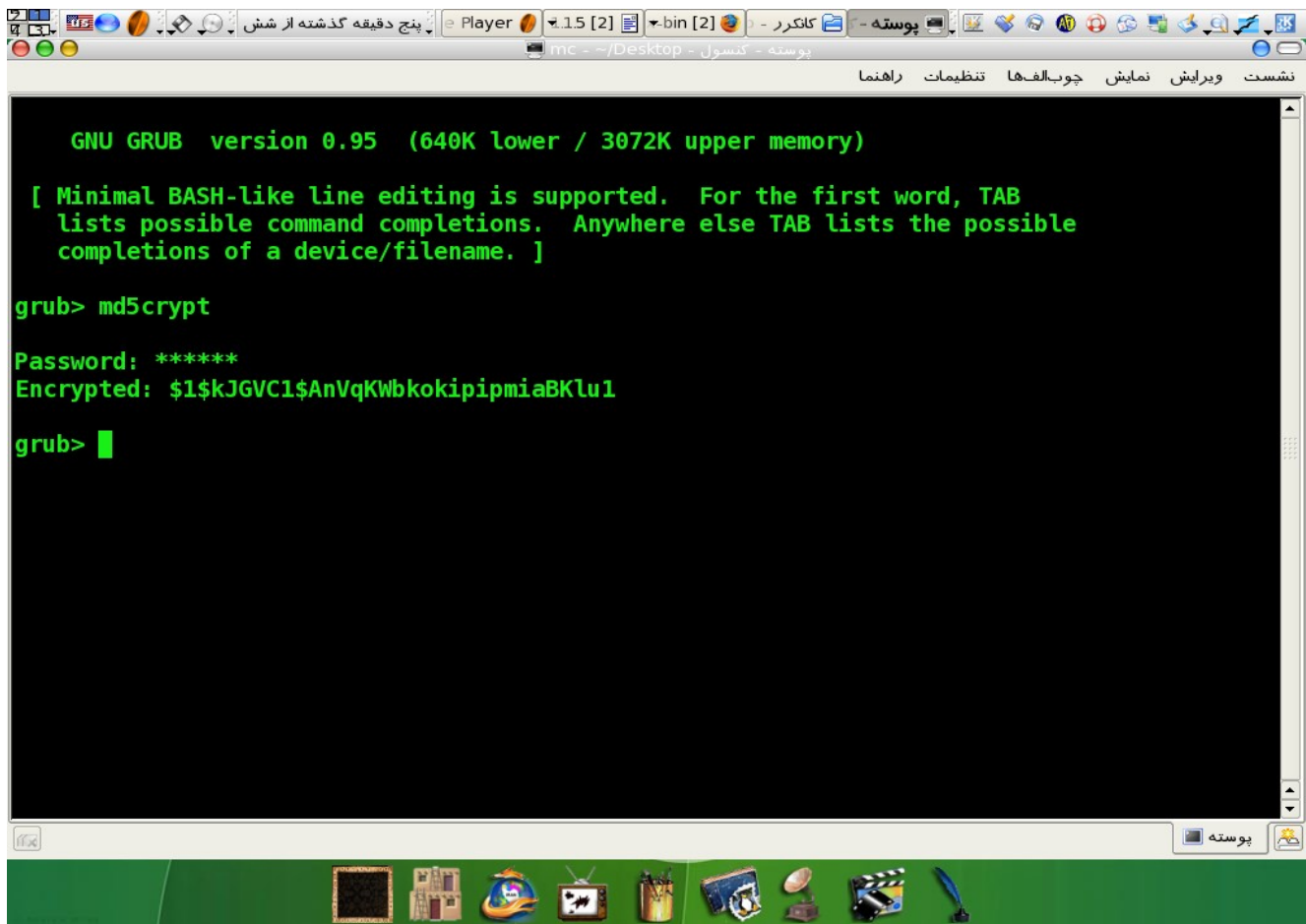
```
gfxmenu (hd0,4)/message
```

```
color white/blue black/light-gray
```

```
default 0
```

```
timeout 8
```

```
password --md5 $1$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```



ایجاد یک کلمه عبور رمزگذاری شده در grub

پس از اعمال این تنظیمات، دستورات مربوط به پوسته GRUB در زمان راه اندازی سیستم، که امکان دسترسی به سیستم فایل‌های موجود را در این مرحله میسر می‌سازد، تنها پس از فشردن کلید P و وارد کردن گذرواژه تعریف شده امکان پذیر خواهد بود.

۴) جهت جلوگیری از بالا آمدن سیستم بوسیله یکی از سیستم عامل‌های موجود در منوی Boot می‌بایست در هر بخش از فایل menu.lst که نمی‌خواهیم بدون وارد کردن گذرواژه راه اندازی شود، عبارت lock را وارد کنیم. بطور مثال داریم:

```
title linux
```

```
kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
```

```
initrd (hd0,4)/initrd
```

```
lock
```

پس از راه اندازی مجدد سیستم و انتخاب یکی از گزینه‌های موجود در منوی راه اندازی (که lock شده‌اند) با پیغام خطای زیر مواجه می‌شویم:

شویم :

Error 32: Must be authenticated

با فشردن کلید Enter و زدن کلید P اعلان دریافت گذرواژه ظاهر میشود که با وارد کردن آن راه اندازی انجام میشود.

نکته : تنظیم گذرواژه در زمان راه اندازی سیستم بر روی GRUB ، موجب غیرفعال شدن صفحه گرافیکی منوی راه اندازی GRUB میشود.

مراحل تهیه یک فلاپی یا CD راه انداز GRUB

برای ساخت یک فلاپی که در سکتور صفر آن GRUB قرار داده شده است و قابلیت راه اندازی یک سیستم توسط آن وجود دارد ، ابتدا فایل‌های مورد نیاز در پروسه راه اندازی سیستم را در فلاپی قرار می دهیم و سپس فرامین زیر را اجرا می کنیم :

```
# fdformat /dev/fd0
```

```
# mkfs -t msdos /dev/fd0
```

```
#mount /dev/fd0 /media/floppy
```

```
#cd /media/floppy
```

```
#mkdir -p boot/grub
```

```
# cp /usr/local/share/grub/i386-pc/stage* /media/floppy/boot/grub
```

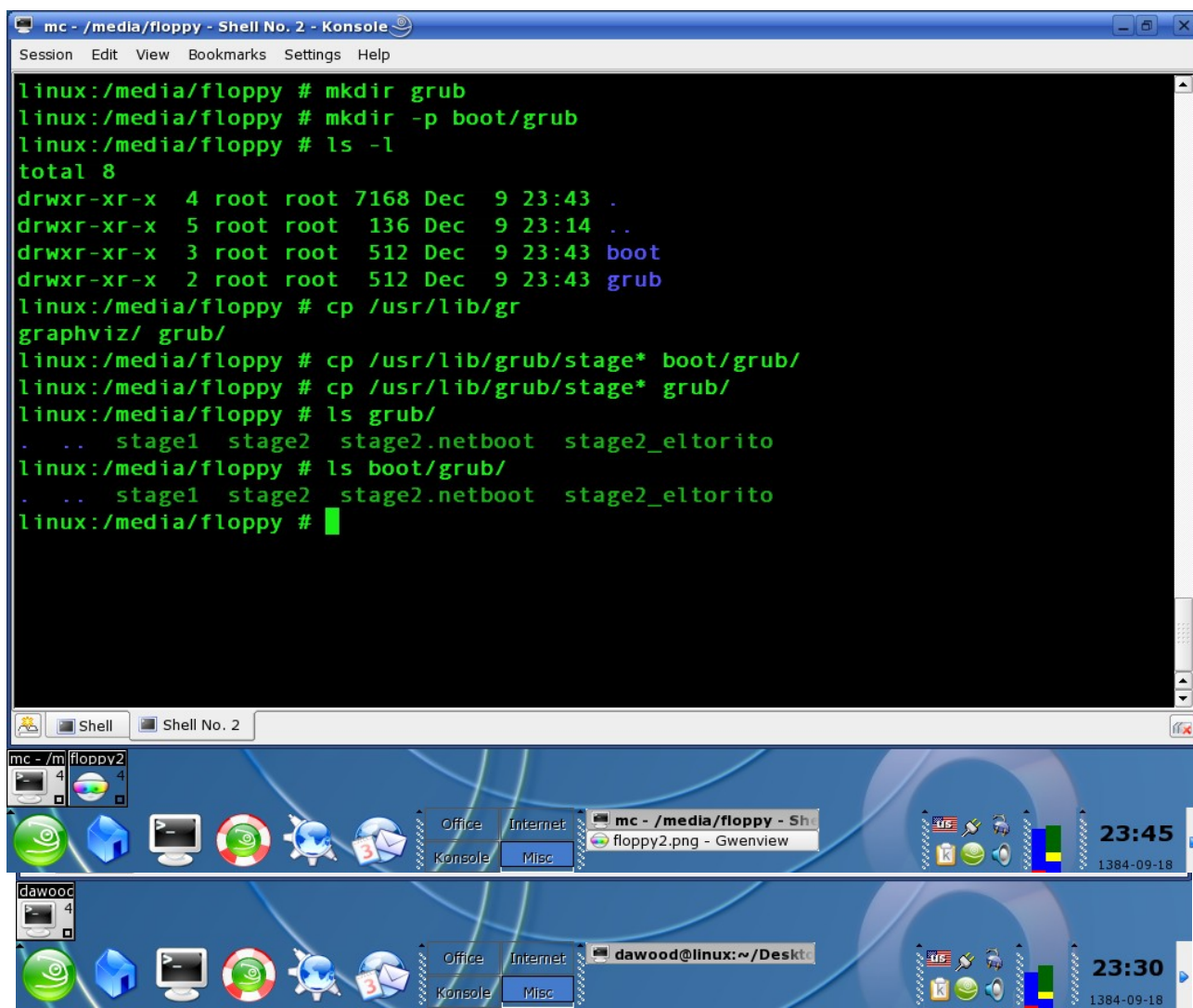
نحوه ساخت یک فلاپی راه انداز در Console

نکته: قبل از انجام عمل format حتما از umount بودن فلاپی درایو مطمئن شوید. در برخی از توزیع ها نیز ممکن است فایل های stage در مسیر مشابهی مانند مسیر یاد شده قرار گیرند. بطور مثال در توزیع SuSE، مسیر مذکور به صورت `usr/lib/grub/stage*` می باشد. پس از کپی کردن فایل های مذکور، فلاپی را umount کرده و سپس جهت نصب GRUB در MBR فلاپی دیسک فرمان grub را در خط فرمان کنسول اجرا می نماییم. با اجرای فرمان مذکور یک برنامه شبیه ساز جهت اجرای فرامین پوسته GRUB اجرا می شود که ۳ فرمان زیر را جهت نصب GRUB بر روی MBR فلاپی دیسک در خط فرمان آن وارد می کنیم:

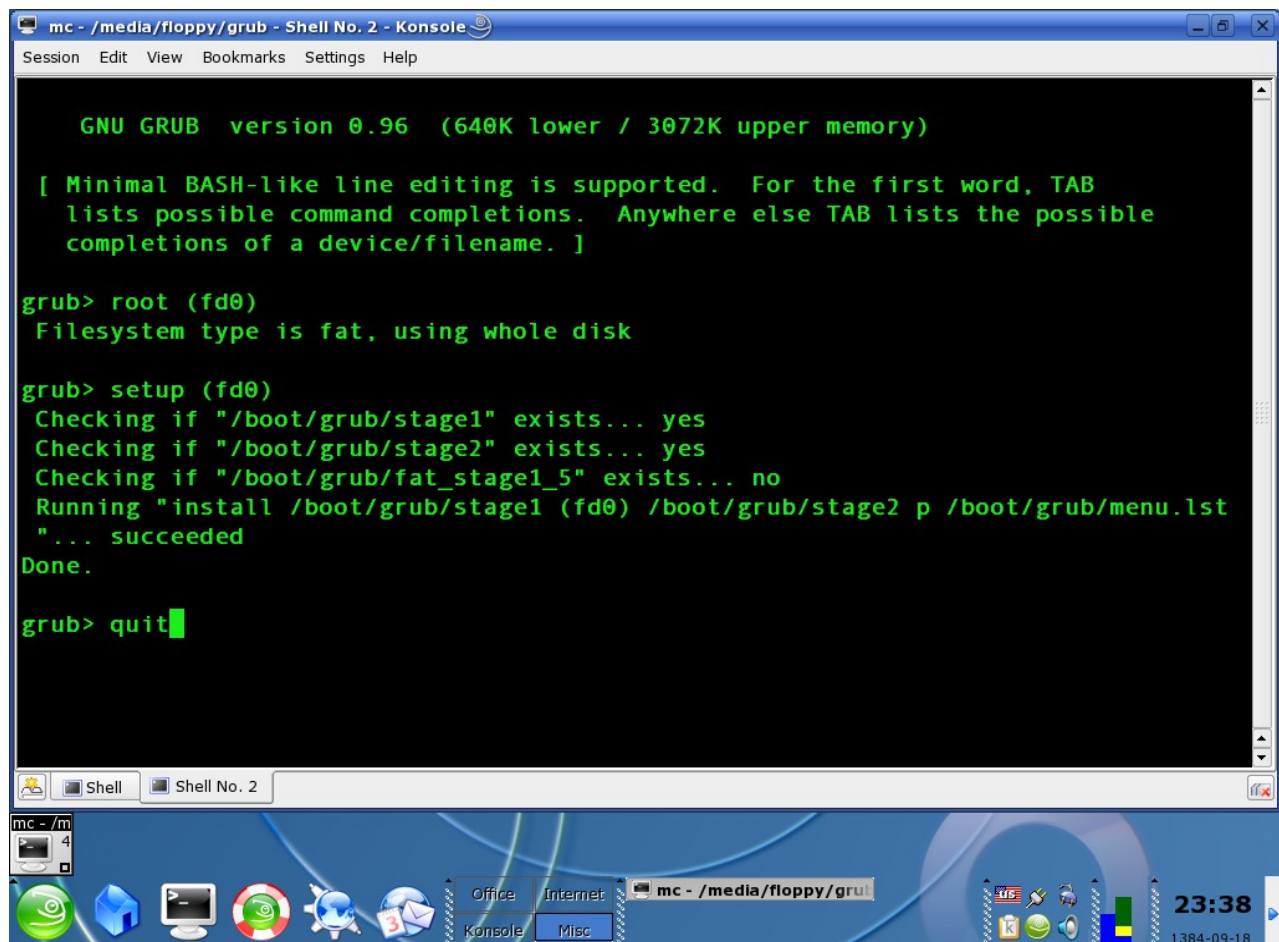
```
grub > root (fd0)
```

```
grub > setup (fd0)
```

```
grub > quit
```



ساخت شاخه های موردنیاز در فلاپی و کپی کردن فایل های مربوط به بارگذار راه انداز در آن



```
mc - /media/floppy/grub - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

GNU GRUB version 0.96 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported. For the first word, TAB
  lists possible command completions. Anywhere else TAB lists the possible
  completions of a device/filename. ]

grub> root (fd0)
Filesystem type is fat, using whole disk

grub> setup (fd0)
Checking if "/boot/grub/stage1" exists... yes
Checking if "/boot/grub/stage2" exists... yes
Checking if "/boot/grub/fat_stage1_5" exists... no
Running "install /boot/grub/stage1 (fd0) /boot/grub/stage2 p /boot/grub/menu.lst
"... succeeded
Done.

grub> quit
```

مراحل اصل ساخت فلاپی راه انداز در محیط نرم افزار grub

خروجی موفقیت آمیز این فرامین ، یک دیسکت جهت راه اندازی سیستم عامل ها می باشد.
برای ساخت CD راه اندازی GRUB نیز عموماً از فایل stage2-eltorito و یک فایل پیکربندی menu.lst استفاده می کنیم و نیازی به
فایل های stage1 و stage2 نمی باشد . حال شاخه ای جهت قرار دادن فایل های مورد نیاز برای نوشتن در CD راه اندازی در مسیر /tmp
ایجاد کرده و فایل stage2-eltorito را در آن کپی می کنیم .

```
$ cd /tmp
```

```
$ mkdir -p iso/boot/grub
```

```
$ cp usr/lib/grub/stage2-eltorito iso/boot/grub
```

همچنین پس از کپی فایل مذکور ، هسته لینوکس (vmlinuz) و فایل‌های initrd و /boot/message را نیز به مسیر tmp/iso/boot کپی می‌نماییم :

```
$ cp /boot/vmlinuz /tmp/iso/boot
```

```
$ cp /boot/initrd /tmp/iso/boot
```

```
$ cp /boot/message /tmp/iso/boot
```

در انتها می‌بایست یک کپی از فایل menu.lst را جهت استفاده از فایل‌های کپی شده در شاخه iso توسط GRUB ویرایش نماییم و در نهایت آنرا به همراه سایر محتویات iso بر روی CD قرار دهیم محتویات فایل menu.lst نیز به شکل زیر تغییر خواهد کرد :

```
gfxmenu (cd) /boot/message
```

```
timeout 5
```

```
default 0
```

```
title linux
```

```
kernel (cd) /boot/vmlinuz root=/dev/hdaX vga=794 resume=/dev/hdaY showopts
```

```
initrd (cd) /boot/initrd
```

جهت ساخت یک فایل boot iso از فایل‌ها و منابع مذکور نیز از فرمان mkisofs به شکل زیر استفاده می‌کنیم :

```
mkisofs -R -b boot/grub/stage2-eltorito -no-emul-boot -boot-load-size 4 -boot-info-table -o grub.iso iso
```

پرسشهای متداول GRUB

در این بخش به بررسی برخی از مشکلاتی که اکثر کاربران در زمان راه اندازی GRUB ممکن است با آنها مواجه شوند می‌پردازیم و راه حل‌های ممکن در هر مورد را ذکر می‌نماییم . اگر سوال شما در این بخش وجود نداشت ، می‌توانید با مراجعه به آدرس زیر و وارد کردن کلمه GRUB به جستجوی آن پرداخته و مطالب بیشتری را در ارتباط با موضوع مورد نظر خود مشاهده نمایید :

<https://portal.suse.com/PM/page/search.pm>

(۱) مشکل نصب Windows پس از نصب GNU/Linux :

از جمله مواردی که اکثر افراد پس از نصب ویندوز در کنار گنو/لینوکس با آن مواجه می‌شوند ، عدم شناسایی گنو/لینوکس در زمان راه

اندازی سیستم است که در این حالت گزینه ای در زمان راه اندازی سیستم برای آن در نظر گرفته نشده است و سیستم مستقیماً وارد محیط ویندوز می شود . علت بروز چنین مشکلی ، حذف شدن بارگذار راه انداز توزیع گنو/لینوکسی شما توسط بارگذار راه انداز ویندوز است. اکثر بارگذار راه اندازهای گنو/لینوکسی پس از نصب ، قابلیت انتخاب و بارگیری سیستم عاملهای میکروسافتی را در زمان راه اندازی سیستم به سایر کاربران ارائه می دهند . این امر بوسیله حفظ و نگهداری آدرس سایر سیستم عاملهای میکروسافتی بر روی دیسک سخت توسط GRUB یا LILO صورت می گیرد . ولی در زمان نصب سیستم عامل های ویندوزی ، محتویات MBR سیستم به طور کامل حذف شده و تنها آدرس سیستم عامل های میکروسافتی در آن قرار می گیرد . در نتیجه سیستم قادر به شناسایی محل سیستم عامل های گنو/لینوکسی و انتخاب و بارگیری آنها در زمان راه اندازی نمی باشد . جهت رفع این مشکل می بایست مجدداً اقدام به نصب بارگذار راه انداز توزیع گنو/لینوکس موجود پرداخته و تنظیمات مورد نیاز آن برای سیستم عامل های میکروسافتی را در صورت لزوم نیز اعمال نماییم . انجام این پروسه در توزیع های مختلف معمولاً به روشهای متنوعی آنهم توسط اولین CD نصب آنها صورت می گیرد . به طور نمونه ، جهت بازسازی مجدد GRUB پس از بروز چنین مشکلی در توزیع SuSE با استفاده از اولین CD آن و انجام مراحل زیر می توان بارگذار راه انداز SuSE را براحتی مجدداً نصب کرد:

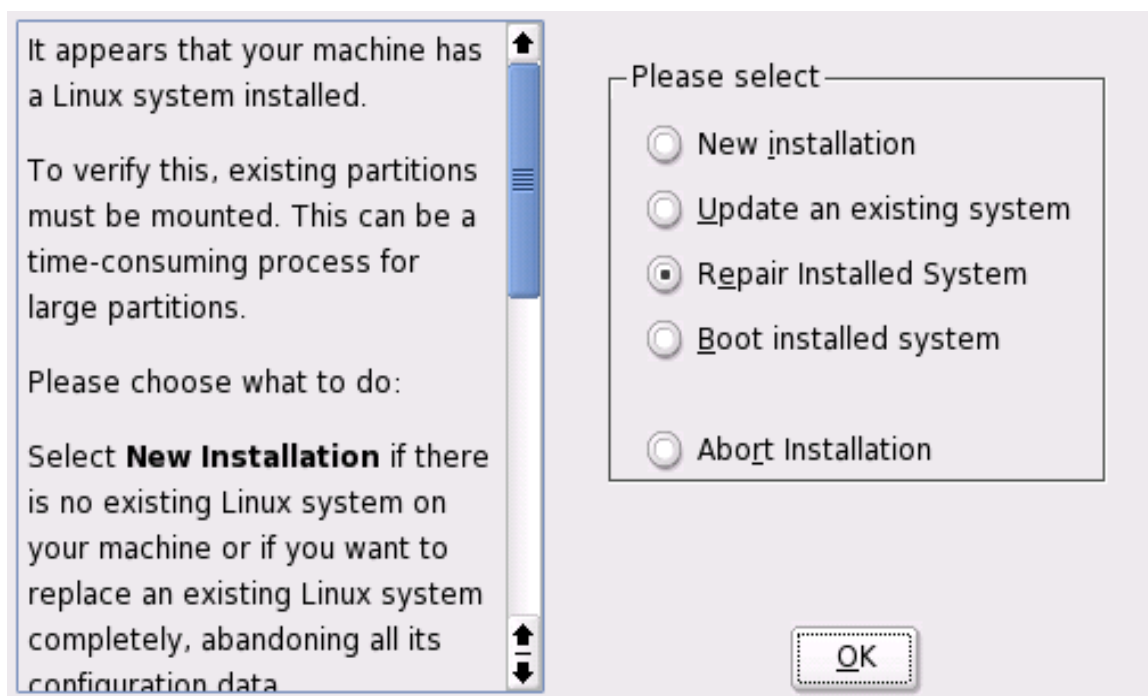
۱ – قرار دادن CD شماره ۱ توزیع SuSE و راه اندازی سیستم با آن

۲ – انتخاب گزینه installation

۳ – انتخاب گزینه Repair installed system پس از انتخاب زبان نصب

۴ – انتخاب گزینه Expert Tools و به دنبال آن Install new boot loader

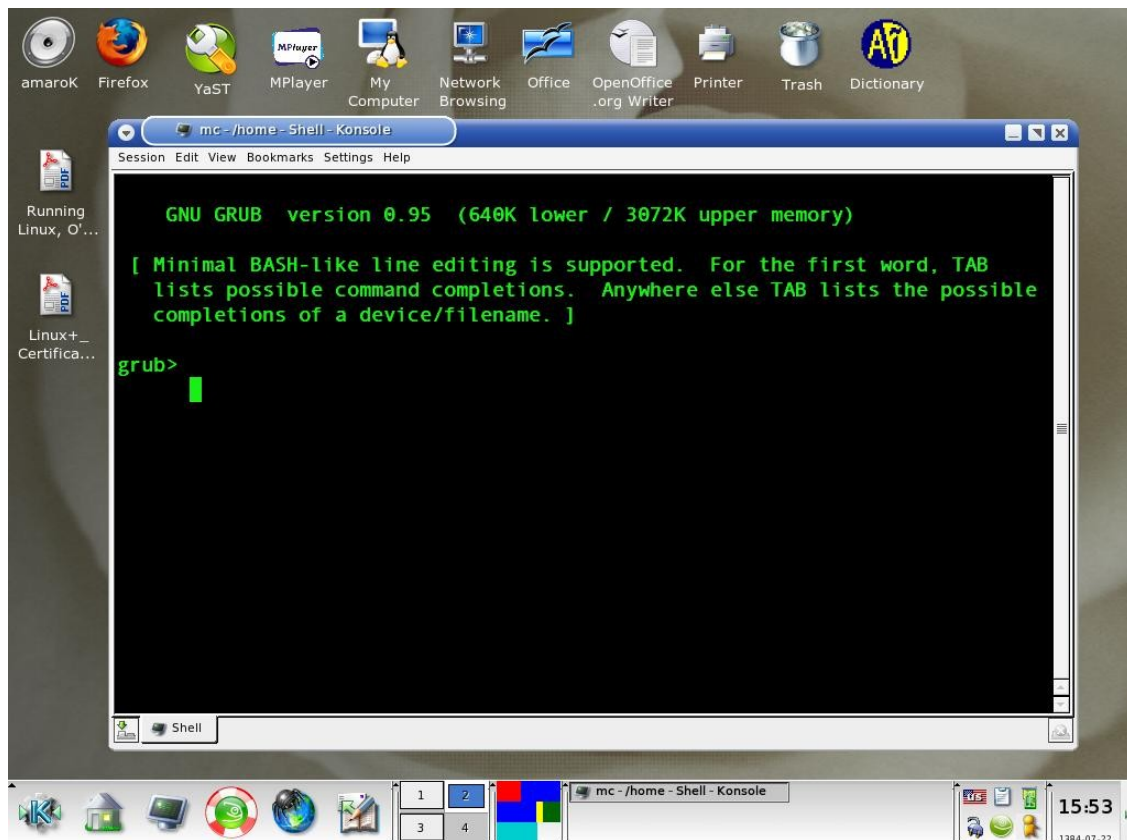
با کلیک بر روی لینک install new boot loader ، نصب مجدد بارگذار راه انداز سیستم آغاز شده و پس از اتمام نصب آن با فشردن کلید Finish سیستم مجدداً راه اندازی شده و این بار از GRUB جهت انتخاب و بارگیری سیستم عامل های موجود در سیستم شامل Windows و سایر توزیع های گنو/لینوکسی استفاده می کند . در دیگر توزیع های لینوکسی مانند Ubuntu , Gentoo , Redhat و ... این رویه در زمان استفاده از CD شماره ۱ و مد گرافیکی قدری تفاوت دارد .



بازسازی بارگذار راه انداز سیستم با استفاده اولین دیسک توزیع SuSE

در حالت کلی هرگاه پس از بروز چنین مشکلی قصد رفع آنرا داشتیم ، می توانیم این کار را در محیط Console یا پوسته ای (shell) که توسط Rescue mode در اولین CD توزیع یا یک Live CD موجود ایجاد شده است، انجام دهیم . بدین منظور ابتدا می بایست در MBR سیستم GRUB را مجددا نصب نماییم که این کار به صورت زیر انجام می گیرد :

grub



```
grub > find /boot/grub/stage1
```

خروجی دستور فوق که به صورت (hdX,Y) بوده و در آن X و Y اعدادی بزرگتر یا مساوی صفر می باشند، به عنوان ورودی به شکل زیر به فرمان setup داده شده و با quit از محیط فرمان grub خارج می شویم .

```
grub > root (hdX,Y)
```

```
grub > setup (hdX)
```

```
grub > quit
```

در صورت عدم استفاده از فرمان root معمولاً می بایست پارتیشنی را که فایل های پیکربندی GRUB در آن قرار گرفته است را mount کرده و عبارات لازم برای انتخاب و بارگیری Windows در زمان راه اندازی سیستم را وارد می نماییم . مثال زیر نمونه ای از این کار را نشان می دهد :

```
#mount /dev/hdx /mnt
```

```
#vi /mnt/boot/grub/menu.lst
```

گزینه های زیر نیز باید به عنوان عبارات لازم جهت راه اندازی ویندوز در فایل menu.lst قرار گیرند :

```
title WindowsXP
```

```
rootnoverify (hd0,3)
```

```
chainloader +1
```

۲) GRUB و سیستم فایل XFS :

سیستم فایل XFS مکانی را برای قرار گرفتن بخش stage1، در بلوک راه انداز سیستم در نظر نمی گیرد. بنابراین نمی توان یک پارتیشن XFS را بعنوان محلی برای قرار گرفتن Boot loader انتخاب کرد. این مشکل را می توان با راحتی با ساخت یک boot partition مجزا با سیستم فایل XFS برطرف نمود.

۳) GRUB و سیستم فایل JFS :

اگرچه از نقطه نظر فنی قرار گرفتن GRUB در پارتیشنی با سیستم فایل JFS امکان پذیر است ، ولی عاری از مشکل نخواهد بود . از اینرو در زمان کار با این سیستم فایل بهتر است یک راه اندازی پارتیشن مجزا (/boot) با سیستم فایل Ext2 یا Ext3 ساخته و GRUB را در این پارتیشن نصب کنیم .

۴) GRUB پیغام خطای "GRUB Geom Error" را نمایش می دهد

GRUB ساختار تمام دیسکهای سخت متصل به سیستم را در زمان راه اندازی چک می کند. در برخی موارد BIOS اطلاعات نامعتبری را برگردانده و پیغام خطای "GRUB Geom Error" را نمایش می دهد. در چنین مواردی می بایست یا GRUB را بروز رسانی کرده و یا از راه انداز LILO استفاده نماییم. همچنین ممکن است این پیغام را زمانی که گنو/لینوکس بر روی دیسک سخت دیگری نصب شده است و این دیسک توسط BIOS تشخیص داده نشده است نیز مشاهده کنیم. سیستم هایی که دارای دیسک سخت IDE و SCSI بوده و stage1 بارگذار راه انداز بدرستی در حافظه قرار می گیرد ولی stage2 آن توسط BIOS پیدا نمی شود با این مشکل مواجه می گردند. این مشکل را نیز می توان با معرفی دیسک سخت جدید در BIOS برطرف کرد .

۵) سیستم هایی که همزمان دارای دیسک IDE و SCSI هستند ، راه اندازی نمی شوند

در زمان نصب گنو/لینوکس ممکن است که مدیر نصب سیستم مانند Yast یا Anaconda روند راه اندازی دیسکهای سخت سیستم را

بدرستی تعیین نکند و شما نیز آنرا در طول نصب اصلاح نکنید . بطور مثال GRUB ممکن است دیسک موجود در مسیر /dev/hda را بعنوان hd0 و دیسک مسیر /dev/sda را بعنوان hd1 در نظر بگیرد، هرچند که رویه راه اندازی در BIOS خلاف این می باشد، یعنی SCSI قبل از IDE قرار می گیرد. برای رفع این مشکل می بایست از فرمان GRUB استفاده کنیم . پس از راه اندازی سیستم باید فایل device.map را جهت شناسایی صحیح دیسکها و نگاشت آنها به مسیر متناظرشان ویرایش نماییم. پس از اینکار بایداسامی ابزارهای موجود در فایل های /boot/grub/menu.lst و /boot/grub/device.map را چک کرده و GRUB را مجدداً توسط فرمان زیر نصب نماییم :

```
#grub -bash > /etc/grub.conf
```

۶) راه اندازی Windows از دیسک سخت دوم

برخی از سیستم عامل ها مانند Windows تنها قادر به راه اندازی از اولین دیسک سخت هستند . اگر چنین سیستم عامل هایی را بر روی دیسکی غیر از اولین دیسک سخت موجود سیستم نصب کردید ، می توانید با یک تغییر منطقی در منوی ورودی سیستم آنرا بدین شیوه اصلاح کنید :...

```
title windows
```

```
map (hd0) (hd1)
```

```
map (hd1) (hd0)
```

```
chainloader(hd1,0)+1
```

در این مثال windows از دومین دیسک سخت راه اندازی می شود که برای انجام این امر موقعیت منطقی دیسکها را با استفاده از map تغییر می دهیم . این تغییرات تأثیری بر منطق حاکم بر منو فایل GRUB ندارند بنابراین دومین دیسک سخت را باید بعنوان chainloader مشخص نماییم.

۷) راه اندازی Windows در صورت غیر قابل دسترس بودن گنو/لینوکس

در مواردی ممکن است برخی از پارتیشن های سیستم عامل گنو/لینوکس موجود در سیستم خصوصاً پارتیشن ریشه دچار مشکل شده و امکان راه اندازی سیستم عامل وجود نداشته باشد . در این حالت ، زمان راه اندازی سیستم عبارت > grub بر روی صفحه نمایش ظاهر شده و عملیات راه اندازی سیستم متوقف می گردد و سیستم تنها منتظر دریافت فرمان از طرف کاربر برای مفسر GRUB خود می باشد . در چنین حالتی اگر دارای سیستم عامل Windows نیز بوده و خواستار راه اندازی سیستم توسط آن هستیم ، کافی است که فرامین زیر را

به ترتیب در خط فرمان grub وارد نماییم :

```
grub > rootnoverify (hd0 , 0)
```

```
grub > makeactive
```

```
grub > chainloader +1
```

```
grub > boot
```

در این بخش به عنوان گزینه makeactive عملکرد سایر فرامین موجود در این قسمت در بخشهای گذشته تشریح شدند . تنها گزینه در جدول پارتیشن سیستم می شود rootnoverify مربوط به پارامتر ورودی فرمان flag جدیدی به کار رفته است که موجب فعال شدن

نکاتی که در عیب یابی GRUB می توانند کارساز باشند :

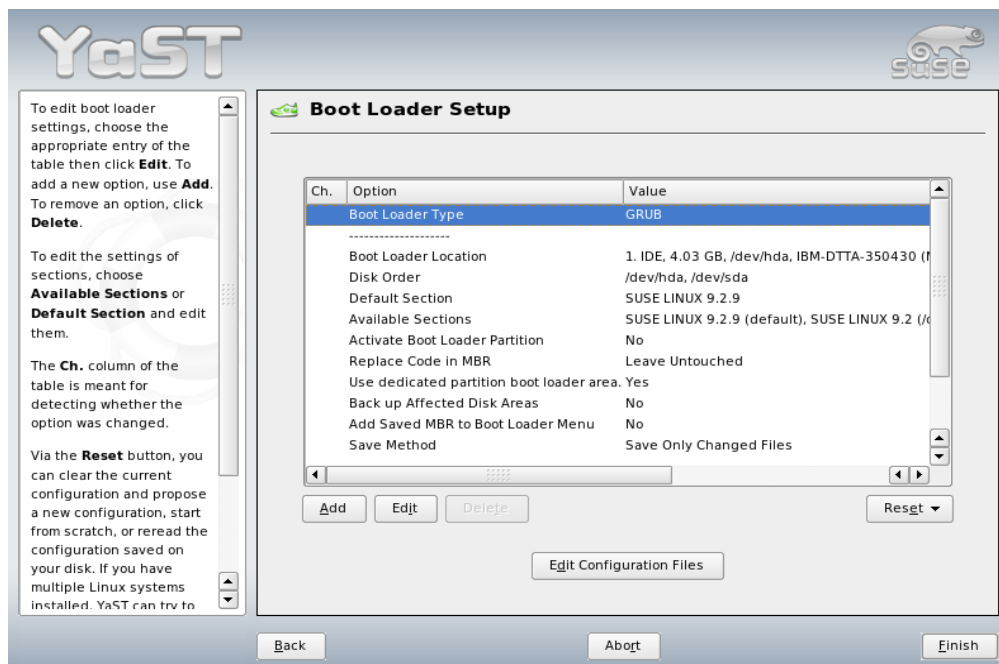
نکته (۱) از فرمان cat جهت مشاهده محتوای فایل ها در GRUB نیز می توان استفاده کرد ، مانند :

```
grub > cat (hd0,2) /etc/fstab
```

نکته (۲) جهت مشاهده پارامترهای kernel در حین استفاده از فرمان grub می توان از عبارت **help kernel** استفاده نماییم .

پیکربندی بارگذار راه انداز بوسیله ابزار گرافیکی Yast

منوی پیکربندی بارگذار راه انداز در ابزار Yast از توزیع SuSE 9.3



ساده ترین راه برای پیکربندی بارگذار راه انداز در توزیع SuSE ، استفاده از ابزار مدیریتی Yast می باشد . برای انجام این امر از بخش Boot loader configuration گزینه System در ابزار Yast می توان استفاده کرد . با ورود به این قسمت ، پیکربندی جاری بارگذار راه انداز شما نمایش داده می شود که در همین جا شما قادر هستید تغییرات لازم را به آن اعمال نمایید .

تشریح اجزاء پیکربندی

همانطور که مشاهده می کنید جدول موجود در شکل فوق شامل ۳ ستون می باشد . جهت افزودن یا ویرایش گزینه های جدید در فایل پیکربندی GRUB از کلیدهای Add و Edit استفاده می کنیم . پارامترهایی که با فشردن کلید Reset تنظیم میگردند به شرح زیر می باشند :

Propose New Configuration

پیکربندی جدیدی را به سیستم عامل ارائه می دهد که توسط آن شما قادر خواهید بود سایر سیستم عاملهای موجود در دیگر پارتیشن ها را به عنوان گزینه ای در زمان راه اندازی داشته باشید . در حقیقت این امر موجب شناسائی و فعالسازی این پارامترها در بارگذار راه انداز گنو/لینوکس می شود .

Start from Scratch

توسط این گزینه می توان تمام اجزای پیکربندی را از ابتدا و به صورت اجزاء جدا از یکدیگر ایجاد کرد .

Reread Configuration from disk

اگر در پیکربندی سیستم تغییراتی را اعمال کرده ایم و از نتایج حاصل از آن راضی نیستیم ، با این گزینه می توانیم پیکربندی جاری سیستم را مجدداً بازبینی نمائیم .

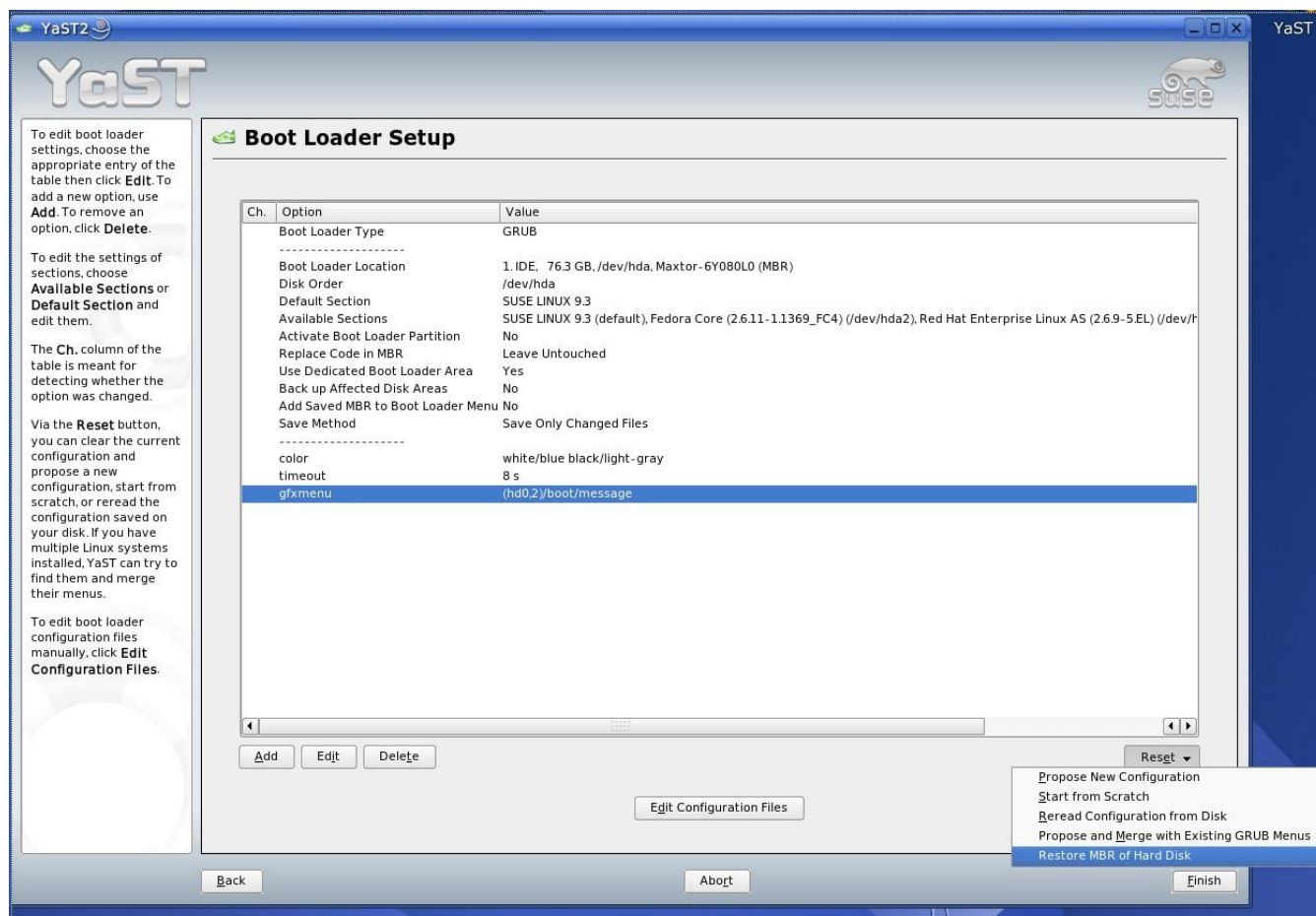
Propose and Merge with Existing GRUB menus

با انتخاب این گزینه در صورت وجود سیستم عاملهای گنو/لینوکسی در کنار سایر سیستم عامل ها بر روی پارتیشن ها می توان گزینه ای را جهت انتخاب سیستم عامل جدید در کنار گزینه های قبلی بارگذار راه انداز سیستم ایجاد نماییم . انجام این کار زمان نسبتاً زیادی را می گیرد و استفاده از آن در LILO مقدور نمی باشد .

Restore MBR of Hard Disk

MBR قبلی سیستم را که از آن پشتیبان تهیه کرده بودیم ، توسط این گزینه می توان بازبینی کرد .

اجزاء تشکیل دهنده گزینه Reset در ابزار پیکربندی Yast

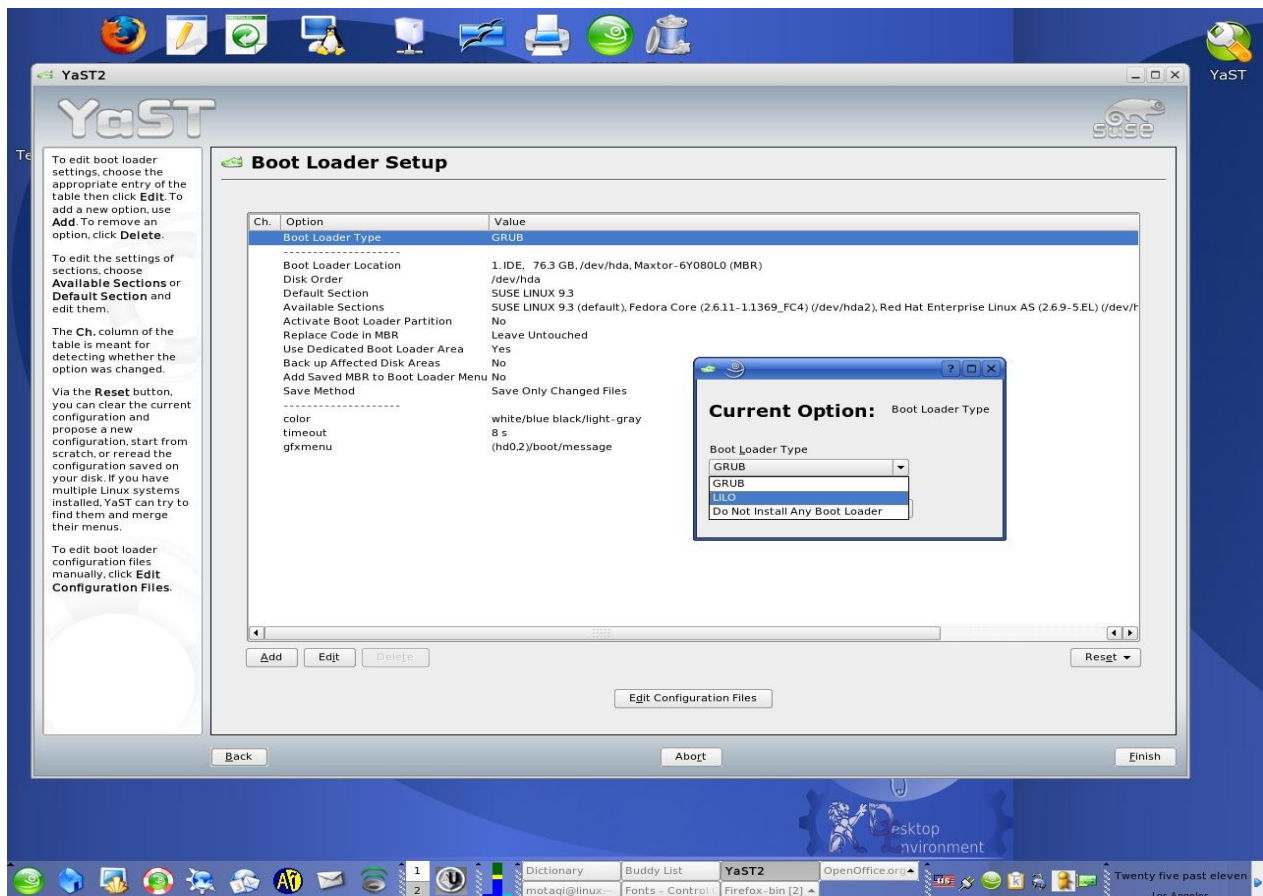


Boot loader Configuration Option

پیکربندی پروسه راه اندازی سیستم توسط GRUB به مراتب ساده تر از ویرایش مستقیم فایل‌های پیکربندی سیستم است . جهت انجام این امر بر روی گزینه Edit کلیک کرده و توسط پنجره ای که باز می شود می توانیم تنظیمات مورد نظرمان را اعمال نماییم . از گزینه های موجود در این لیست ، جهت ایجاد تغییرات در بخشهای مختلف بارگذار راه انداز استفاده می کنیم . برخی از این گزینه ها به شرح زیر می باشند :

Boot loader type

جهت انتخاب نوع بارگذار راه انداز سیستم (GRUB or LILO) از این گزینه استفاده می کنیم . با استفاده از گزینه های موجود می توان پیکربندی بارگذار راه انداز موجود مانند GRUB را به پیکربندی متناظرش در LILO تبدیل نمود ، اگرچه برخی از تنظیمات در این بین ممکن است بدلیل عدم داشتن معادل برای آنها نادیده گرفته شوند . اگر قصد بازگشت به پیکربندی اصلی بارگذار راه انداز سیستم را داشتیم ، می توانیم پیکربندی آنرا بوسیله گزینه های قبلی بازیابی نماییم . تمام این قابلیتها در صورت بسته نبودن پیمانه (Module) بارگذار راه انداز در سیستم قابل استفاده می باشند .



تعیین نوع بارگذار راه انداز سیستم توسط Yast در SuSE 9.3

Boot Loader Location

مکانی که قصد نصب کردن بارگذار راه انداز را در آنجا داریم ، به کمک این پارامتر تعیین می نماییم که می تواند یکی از گزینه های زیر باشد:

- Master Boot Record سیستم

- سکتور پارتیشن راه اندازی (در صورت وجود)

- سکتور پارتیشن ریشه و یا یک فلاپی دیسک.

جهت انتخاب دیگر گزینه ها برای نصب بارگذار راه انداز در مکانی دیگر ، از کلید other استفاده می نماییم .

Disk Order

اگر سیستم ما دارای بیش از یک دیسک سخت باشد، رویه و ترتیب راه اندازی دیسکهای تعریف شده در BIOS را توسط این گزینه انجام می دهیم .

Default Section

این گزینه هسته یا سیستم عاملی را که در زمان راه اندازی باید به عنوان گزینه پیش فرض در نظر گرفته شود تعیین می کند . سیستم عامل یا هسته مورد نظر ، پس از به پایان رسیدن زمان timeout ، به صورت خودکار راه اندازی خواهد شد. در این منو نیز قادر خواهیم بود لیستی از تمام اجزاء منوی راه اندازی سیستم را مشاهده نموده و در صورت لزوم آنها را بوسیله گزینه Edit اصلاح نماییم . با انتخاب هر کدام از سیستم عاملهای دیگر موجود در این لیست و فشردن گزینه Set as Default می توانیم آنها را به عنوان گزینه پیش فرض انتخاب نماییم .

Make Boot Loader Partition Active

از این گزینه جهت فعالسازی پارتیشنی که سکتور راه اندازی حاوی بارگذار راه انداز را دارا می باشد ، استفاده می کنیم . پارتیشن مذکور را مستقل از فایلهای مورد نیاز بارگذار راه انداز که عموماً در مسیر boot/ و یا / قرار می گیرند ، انتخاب می کنیم .

Replace Code in MBR

اگر نصب GRUB در MBR سیستم دچار مشکل شده است و یا شما در حال نصب GRUB بر روی یک دیسک سخت جدید هستید، با استفاده از این گزینه می توان کد عمومی راه اندازی سیستم را در MBR بازنویسی کرد.

Back up Files and Parts of Hard Disks

بخشهایی از دیسک سخت که داده های آنها تغییر یافته است را پیدا کرده و از آنها پشتیبان تهیه می کند .

Add Saved MBR to Boot Loader Menu

MBR ذخیره شده در سیستم را ، به عنوان گزینه ای ، در داخل منوی بارگذار راه انداز قرار می دهد .

با استفاده از پارامتر Time out نیز می توان مدت زمانی که سیستم منتظر دریافت یک ورودی از صفحه کلید جهت انتخاب یکی از گزینه های راه اندازی سیستم باشد را انتخاب کرد .

حذف بارگذار راه انداز گنو/لینوکس

در گنو/لینوکس SuSE توسط ابزار Yast قابلیت حذف بارگذار راه انداز جاری سیستم و بازیابی مقدار قبلی MBR سیستم را دارا هستیم . در طول رویه نصب ، Yast به شکل خودکار یک کپی پشتیبان از MBR اصلی سیستم تهیه کرده و در صورت نیاز در بخش system ابزار Yast و انتخاب گزینه Reset MBR of HDD از قسمت Reset می توان براحتی GRUB سیستم را uninstall نمود . همچنین در مواردی

که بخواهیم بارگذار راه انداز گنو/لینوکس ما به طور کامل از MBR سیستم حذف شده و بارگذار Windows را جایگزین آن نماییم ، می توانیم با استفاده از یک فلای راه اندازی Dos/Windows سیستم را راه اندازی کرده و سپس در خط فرمان ، دستور زیر را جهت پاک کردن محتویات MBR تایپ کنیم :

A : \> fdisk /mbr

حال برای قرار دادن محتویات بارگذار راه انداز Microsoft بر روی MBR فرمان : sys c: را از روی دیسک راه اندازی اجرا می نماییم.

بخش سوم

مقدمه ای بر LILO

بارگذار راه انداز lilo برگرفته از Linux loader (بارگذار لینوکس) بوده و توسط جان کافمن توسعه داده شد. این بارگذار تا قبل از نسخه ۷.۲ توزیع Redhat، در این توزیع و سایر توزیع های مشهور گنو/لینوکسی به شکل پیش فرض مورد استفاده قرار می گرفت. هم اکنون lilo به هیچ سیستم خاصی وابسته نبوده و قادر است هسته های مختلف گنو/لینوکس را از دیسکهای سخت و یا فلاپی دیسک بارگیری کند و حتی به عنوان یک مدیر راه انداز برای سایر سیستم عاملها مانند OS/2, Windows, Dos و Scounix بکار رود.

همچنین lilo قادر است تا ۱۶ تصویر مختلف از هسته را در زمان راه اندازی سیستم تشخیص داده و یکی از آنها را انتخاب و بارگیری نماید. بارگذار lilo توسط فایل lilo.conf موجود در مسیر /etc/ پیکربندی می شود. اصول کار با lilo و پیکربندی آن بسیار ساده است، ولی

قدرت اصلی آن متکی به گزینه های متعدد موجود در فایل lilo.conf می باشد که در صورت نیاز از هر کدام از این گزینه ها می توان استفاده کرد .

اجزا تشکیل دهنده LILO

بارگذار lilo متشکل از چندین فایل و برنامه می باشد که به شرح ذیل می باشند :

۱- نصب کننده نقشه (map installer) :

برنامه ای است که با اجرای آن تمام فایل های متعلق به lilo در جای مناسب خود قرار گرفته و مکان اطلاعات و فایل های مورد نیاز در زمان راه اندازی سیستم را نیز ثبت می کند . این برنامه معمولاً در فایل /sbin/lilo قرار می گیرد . با هر بار اعمال تغییرات در ساختار lilo این فایل می بایست اجرا شود تا تغییرات جدید به سیستم اعمال گردد .

۲- بارگذار راه انداز سیستم (Boot loader) :

این بخش از lilo توسط BIOS بارگیری شده و وظیفه بارگذاری هسته ها یا سکتورهای راه انداز دیگر سیستم عامل را بر عهده دارد . همچنین این قسمت خط فرمان ساده ای ر جهت محاوره با سیستم و بکاربردن گزینه های مختلف جهت راه اندازی سیستم عامل های موجود را در اختیار کاربر قرار می دهد .

۳- فایل های متفرقه (Various files) :

این فایل ها حاوی اطلاعاتی هستند که lilo در زمان راه اندازی سیستم از آنها استفاده می کند و معمولاً در مسیر /boot قرار می گیرند . مهمترین فایل موجود در این مسیر فایل نقشه (/boot/map) می باشد که برنامه نصب کننده نقشه به کمک آن مکان هسته های موجود در سیستم را ثبت می کند . در حالت عادی LILO قادر به خواندن فایل سیستم های موجود نمی باشد . برنامه نصب کننده نقشه از هسته سیستم عامل جهت پیدا کردن محل فایل های موجود در سیستم و شناسایی فایل سیستم ها کمک می گیرد و اطلاعات لازم را بوسیله آن کسب می کند .

اجزایی از سیستم که بارگذار lilo عموماً از آنها استفاده می کند عبارتند از :

۱- پارتیشن فایل سیستم ریشه : این پارتیشن از دو جنبه برای lilo دارای اهمیت است : اولاً lilo در برخی موارد باید به هسته بگوید که کجا بدنبال این پارتیشن بگردد . ثانیاً این مکان از دیسک سخت عموماً محلی برای قرار گرفتن سایر اجزا تشکیل دهنده lilo است ، که شامل /boot ، سکتور راه انداز و هسته های موجود می باشد .

۲- سکتور راه انداز : همانطور که در ابتدای کتاب مطرح شد این بخش از دیسک حاوی اولین بخش بارگذار راه انداز سیستم که در اینجا lilo می باشد است . این بخش وظیفه بارگیری دومین جزء بارگذار سیستم را برعهده دارد که اندازه آن از اولین جزء بزرگتر می باشد . هر دو جزء غالباً در فایل boot/boot.b قرار می گیرند .

۳- هسته سیستم عامل : بارگیری و اجرای هسته توسط بارگذار راه انداز سیستم است . هسته ها نیز عموماً در مسیر boot/ از پارتیشن ریشه سیستم قرار می گیرند . در نظر گرفتن این نکته که اکثر فایل های مورد نیاز lilo در زمان راه اندازی سیستم، می بایست توسط BIOS نیز قابل دسترسی باشند، در بررسی سایر موارد و مشکلات احتمالی سیستم دارای اهمیت زیادی بوده و برای ما کارگشا خواهد بود .

فایل lilo.conf و فرمان lilo

جهت راه اندازی صحیح lilo ابتدا می بایست در فایل پیکربندی آن (/etc/lilo.conf) تنظیمات لازم را اعمال کرده و سپس تحت عنوان کاربر ریشه فرمان lilo را اجرا نماییم . اعمال هرگونه تنظیمات اشتباه در فایل پیکربندی lilo و بازنویسی MBR سیستم توسط آن می تواند منجر به ایجاد مشکل برای شما و عدم راه اندازی مجدد سیستم گردد ، از این رو باید علاوه بر داشتن آشنایی کافی با اجزا فایل در زمان تنظیم آن، دقت لازم را در اعمال تغییرات نیز مدنظر قرار دهیم .

جهت آشنایی با سایر پارامترهای اساسی فایل lilo.conf مثالی از این فایل را در زیر مورد بررسی قرار می دهیم که شامل پیکربندی راه اندازی lilo برای سیستمی با دو پارتیشن لینوکسی و یک پارتیشن راه انداز ویندوزی است .

```
boot = /dev/hda
prompt
timeout = 50
compact
vga = extended
lock
password = JANJ”)wo
restricted
append = “ether =9,0x300,0xd0000,0xd4000,etho hisax = 1,3,5,0xd8000,0xd80,hisax”
image = /boot/vmlinuz-2.2.17
label = linux
root = /dev/hda5
read- only
image = /boot/vmlinuz-2.0.38
```

label = linux-old

root = /dev/hda6

read-only

other = /dev/hda2

label = windows

table = /dev/hda

اجرای فرمان lilo موجب نصب راه انداز سیستم در MBR می گردد که از طریق فایل boot/map/ خواهد بود محل قرار گرفتن فایل

vmlinux7-2.2.17 را پیدا کند . خروجی اجرای فرمان lilo به شکل زیر می باشد :

root@linux#lilo

Added linux*

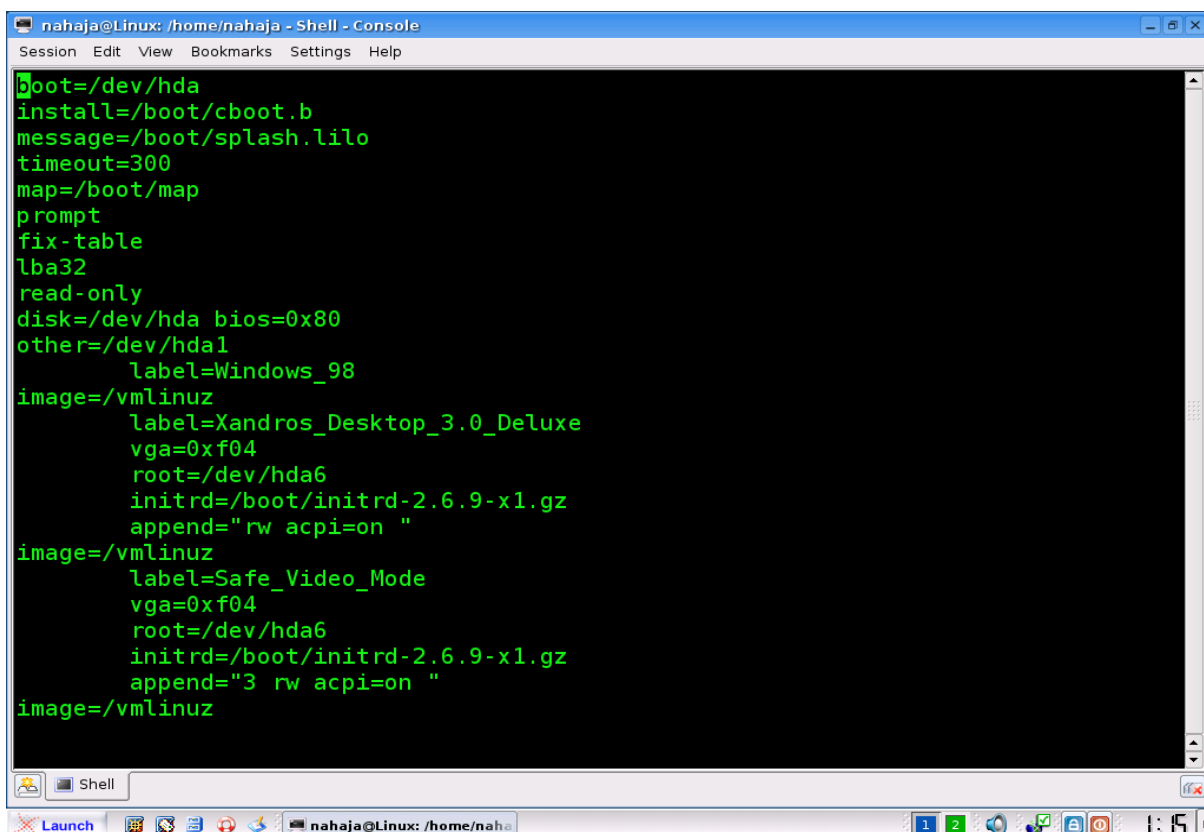
Added linux-old

Added windows

همچنین فرمان lilo از MBR موجود سیستم یک پشتیبان تهیه کرده و آنرا در مسیر /boot/ تحت یک فایل با عنوانی مانند boot.0300

قرار می دهد .

مثال دیگری از یک فایل **/etc/lilo.conf**

A screenshot of a terminal window titled "nahaja@Linux: /home/nahaja - Shell - Console". The window displays the configuration for the lilo.conf file. The configuration includes settings for the boot device (hda), the boot loader (cboot.b), the splash screen (splash.lilo), the timeout (300), the boot map (map), the prompt, the fix-table, the lba32, the read-only flag, the disk (hda) with bios=0x80, the other device (hda1) with label=Windows_98, the image (vmlinuz) with label=Xandros_Desktop_3.0_Deluxe, vga=0xf04, root=/dev/hda6, initrd=/boot/initrd-2.6.9-x1.gz, and append="rw acpi=on ". There are three image entries in total, each with a different label and append string. The terminal window has a menu bar with Session, Edit, View, Bookmarks, Settings, and Help. The status bar at the bottom shows the user's name (nahaja@Linux) and the current directory (/home/nahaja).

```
boot=/dev/hda
install=/boot/cboot.b
message=/boot/splash.lilo
timeout=300
map=/boot/map
prompt
fix-table
lba32
read-only
disk=/dev/hda bios=0x80
other=/dev/hda1
    label=Windows_98
image=/vmlinuz
    label=Xandros_Desktop_3.0_Deluxe
    vga=0xf04
    root=/dev/hda6
    initrd=/boot/initrd-2.6.9-x1.gz
    append="rw acpi=on "
image=/vmlinuz
    label=Safe_Video_Mode
    vga=0xf04
    root=/dev/hda6
    initrd=/boot/initrd-2.6.9-x1.gz
    append="3 rw acpi=on "
image=/vmlinuz
```

- حال به بررسی اجزا موجود در فایل lilo.conf می پردازیم :

boot : وسیله ای را که سیستم از آن راه اندازی می شود مشخص می کند که معمولاً /dev/sda یا /dev/hda است .

prompt : در زمان راه اندازی یک اعلان را به کاربر نشان می دهد که کاربر قادر خواهد بود توسط آن سیستم عامل مورد نظر خود را جهت راه اندازی انتخاب کند .

timeout : مدت زمان نمایش اعلان در هنگام راه اندازی سیستم را مشخص می کند که بر حسب دهم ثانیه است .

Compact : این گزینه بارگیری هسته سیستم را تسریع می کند که این کار از طریق خواندن سکتورهای همجوار به شکل همزمان انجام می گیرد .

vga : با این گزینه وضوح مد متنی صفحه را در زمان راه اندازی سیستم مشخص می کنیم که در اینجا ۸۰*۵۰ تنظیم شده است. این پارامتر ممکن است بوسیله برنامه زمان راه اندازی مقدارش تغییر پیدا کرده و همان مقدار ۸۰*۲۵ شود .

lock : با استفاده از این گزینه آخرین سیستم عامل راه اندازی شده سیستم، بعنوان گزینه پیش فرض انتخاب میگردد .

password : در زمان راه اندازی از کاربر درخواست کلمه عبور می کند .

restricted : هر گاه کاربری قصد اعمال پارامترهای خاصی را به سیستم در زمان نمایش اعلان lilo داشته باشد و بخواهد از این طریق تغییراتی را در روند راه اندازی یا پیکربندی سیستم اعمال کند، از این گزینه جهت دریافت پسورد از وی برای این کار استفاده می شود .

append : گزینه های موجود برای راه اندازی هسته سیستم عامل را در زمان بالا آمدن سیستم مشخص میکند.

image : جهت مشخص کردن هر یک از هسته های موجود گنو/لینوکس در سیستم بکار می رود .

label : این گزینه متنی را که در زمان راه اندازی سیستم برای انتخاب یک سیستم عامل نمایش داده می شود، مشخص می کند .

root : فایل سیستم ریشه را که جهت اتصال هسته سیستم عامل بکار می رود، تعیین می کند .

read-only : وضعیت فایل سیستم ریشه را در زمان راه اندازی اولیه سیستم مشخص می کند که در اینجا بصورت فقط -خواندنی در اختیار کاربر قرار می گیرد . با استفاده از عبارت read-write این پارامتر به وضعیت خواندنی -نوشتنی تغییر می یابد .

other : جهت مشخص کردن دیگر سیستم عامل های موجود برای راه اندازی سیستم بکار می رود .

table : اطلاعات مربوط به جدول پارتیشن دیسک سخت سیستم باید به سکتور راه انداز سیستم عامل تحویل داده شود که این کار از طریق پارامتر مذکور صورت می گیرد .

- در مثال فوق سایر پارتیشنهای دیسک سیستم دارای مشخصات زیر می باشند :

`/dev/hda1` : یک پارتیشن ext2 که به عنوان `/boot` به سیستم متصل شده است .

`/dev/hda2` : پارتیشن windows

`/dev/hda3` : پارتیشن توسعه یافته سیستم (جهت ساخت پارتیشن های منطقی)

`/dev/hda4` : پارتیشن پایه استفاده نشده سیستم

`/dev/hda5` : فایل سیستم ریشه (پارتیشن ریشه)

`/dev/hda6` : دومین فایل سیستم ریشه که حاوی یک توزیع گنو/لینوکس قدیمی تر می باشد .

`/dev/hda?` : پارتیشن هایی مثل `home`, `swap` و دیگر پارتیشن ها را مشخص می کند .

مثال فوق نمونه ای ساده و نسبتاً معمول از یک فایل پیکربندی `lilo` بود که گزینه های لازم جهت راه اندازی سیستمی با سه سیستم عامل را فراهم می کند . جهت مشاهده سایر پارامترهای موجود جهت تنظیم جامع تر فایل `lilo.conf` تنها کافی است که `man page` های فایل `lilo.conf` را با استفاده از فرمان زیر مشاهده کنید :

`man lilo.conf $`

خروجی این فرمان بالغ بر ۲۰ صفحه اطلاعات راجع به عناصر تشکیل دهنده `lilo` و فایل پیکربندی آن می باشد، که بعنوان یکی از اساسی ترین مراجع آن می تواند مورد استفاده قرار گیرد .

اجرای برنامه LILO

پس از پیکربندی فایل `lilo.conf` بر اساس نیازمندیهای موجود، توسط برنامه `lilo` باید بارگذار را در MBR سیستم نصب نمائیم که عموماً به شکل زیر انجام می شود :

`#!/sbin/lilo`

```
nahaja@XANEWAMEQ31: /boot - Shell - Console
Session Edit View Bookmarks Settings Help

Linux:/# cd /etc/
Linux:/etc# ls -l lilo*
-rw-r--r-- 1 root root 812 2005-09-29 04:59 lilo.conf
-rw-r--r-- 1 root root 812 2005-08-23 03:16 lilo.conf~
-rw-r--r-- 1 root root 515 2005-10-01 09:30 lilop.conf
-rw-r--r-- 1 root root 515 2005-08-23 03:16 lilop.conf~
Linux:/etc# lilo
Added Windows_98 *
Added Xandros_Desktop_3.0_Deluxe
Added Safe_Video_Mode
Added Configure_(Expert)
Added SuSE_Linux_on_hda9
Linux:/etc#
```

نحوه بازنویسی

MBR با استفاده از فرمان lilo

دستور فوق شامل گزینه های فراوانی برای اجرا می باشد که بر اساس شرایط و نیاز موجود می توان از آنها استفاده کرد .
به طور مثال یکی از این گزینه ها که معمولاً در حد اشکال زدایی از سیستم (Rescue Mode) کاربرد دارد، گزینه -r می باشد، که در واقع به سیستم اعلام می کند تا سیستم فایل جاری را به سیستم فایل مشخص شده در دستور تغییر داده و سپس فرمان lilo را اجرا نماید . برای نمونه فرض کنید در مد اشکال زدایی (Rescue Mode) سیستم را راه اندازی کرده و پارتیشن /dev/hda1 را به شاخه /mnt/tmp متصل نموده ایم . حال فایل lilo.conf را اصلاح کرده و اشکالات آنرا برطرف می کنیم . اگر پس از انجام اصلاحات برای اعمال پیکربندی جدید در MBR سیستم از فرمان /sbin/lilo استفاده کنیم، سیستم به دنبال فایل lilo.conf در مسیر /etc می گردد، در حالیکه این فایل هم اکنون در /mnt/tmp/etc قرار دارد . بنابراین جهت حل این مشکل و اعلام محل دقیق فایل پیکربندی بارگذار راه انداز سیستم از فرمان lilo به شکل زیر استفاده کنیم :

#/sbin/lilo -r /mnt/tmp

سایر گزینه های موجود در فرمان lilo را با مراجعه به manual های موجود در سیستم می توان مشاهده نمود که غالباً از طریق فرمان

man lilo قابل دسترسی می باشد .

اجرای برنامه LILO

یکی از روشهایی که عموماً جهت بازیابی سیستم های آسیب دیده بکارگرفته می شود، استفاده از فلاپی دیسکت راه انداز است. این گونه مشکلات معمولاً بدلیل از بین رفتن سکتور راه انداز سیستم، بروز مشکل در عملکرد هت دیسک سخت و یا نصب و بازنویسی یک هسته جدید بر روی هسته قدیم سیستم و اجرا نکردن فرمان lilo ، رخ می دهند.

استفاده از فلاپی دیسک ها درچنین شرایطی به دو شکل صورت می گیرد. یکی زمانی است که قصد داریم بوسیله فلاپی دیسک درزمان راه اندازی سیستم پارامترهایی را به هسته ارسال کنیم و از این طریق عملیات بارگیری هسته سیستم عامل را انجام دهیم. دیگری زمانی است که سایر بارگذارهای راه انداز موجود بر روی دیسک مانند GRUB یا LILO نصب شده در سیستم ، قادر به بارگیری هسته نمی باشند.

هرگاه بخواهیم از روی یک فلاپی و با استفاده از LILO سیستم را راه اندازی نماییم کافی است که در خط مربوط به گزینه boot فایل lilo.conf، مسیرنوشتن تصویر lilo را /dev/fd0 معرفی نماییم. اجرای فرمان /sbin/lilo پس ازاعمال این تنظیم برای شما یک دیسک راه انداز lilo می سازد که حاوی اطلاعاتی است که عموماً در MBR دیسک سخت جهت راه اندازی سیستم قرار می گیرد. اکثر توزیع های گنو/ لینوکسی ابزار ساده تری به نام mkbootdisk را جهت این امر در اختیار کاربران قرار می دهند، که معمولاً درشاخه /sbin قرارگرفته است.

هردیسک راه اندازکه توسط mkbootdisk ساخته شود حاوی تصویری ازهسته، یک فایل initrd.img، یک شاخه boot/ حاوی دومین بخش بارگذار راه انداز و نقشه سیستم، یک شاخه /dev شامل فلاپی دیسک (fd0) و یک فایل سیستم ریشه و درنهایت یک شاخه etc/ دارای فایل lilo.conf می باشد. زمانیکه بوسیله دیسک راه انداز LILO سیستم را راه اندازی می کنیم، یک اعلان LILO در اختیار ما قرارمی گیرد که بوسیله آن می توانیم اطلاعاتی ازسیستم را که نیازداریم بدست آورده و یا فرامین لازم جهت بازیابی سیستم را وارد نماییم. ساده ترین روش جهت ساخت یک دیسک راه انداز با مشخصاتی که عنوان شد، استفاده افرمان mkbootdisk به شکل زیر می باشد که مقدار درج شده در جلوی آن بیانگر نسخه هسته سیستم عامل می باشد:

#/sbin/mkbootdisk 2.2.16-22

ساخت یک دیسک راه انداز بدون استفاده از بارگذار راه انداز

باکپی کردن هسته بر روی فلاپی دیسک با استفاده افرمان dd نیز می توان یک دیسک راه انداز مستقل از بارگذارراه انداز موجود سیستم ایجاد کرد. به طورمثال اگرهسته سیستم ما فایل vmlinuz-2.4.7-10 باشد، به این شکل عمل می کنیم :

#dd if=/boot/vmlinuz-2.4.7-10 of=/dev/fd0

سپس با استفاده از فرمان rdev به هسته موجود در روی فلاپی دیسک پارتیشن ریشه سیستم را معرفی می کنیم. فرمان rdev عموماً جهت تنظیم کردن تصویر وسیله ریشه (پارتیشن ریشه) ما بکار می رود و به ندرت در مواردی جهت تنظیم کردن پارتیشن swap، اندازه حافظه RAM و یا مد گرافیکی VGA بکار گرفته می شود. به طور مثال فرمان زیر اولین پارتیشن از دومین دیسک سخت سیستم را به عنوان پارتیشن ریشه سیستم تعریف می کند:

#rdev /dev/fd0 /dev/hdb1

اطلاعات کامل تر در رابطه با دستور rdev نیز از طریق مطالعه manual های آن نیز قابل دسترسی می باشد.

وجود یا عدم وجود یک بارگذار راه انداز، تأثیری در رویه راه اندازی سیستم توسط فلاپی دیسک ندارد، بلکه تنها تفاوتی که در این دو روش وجود دارد این است که در متد دوم، برنامه راه انداز از سکتور راه انداز موجود در فلاپی دیسک به جای MBR اولین دیسک سخت سیستم استفاده می کند. همچنین اگر از یک فلاپی دیسک فاقد بارگذار راه انداز استفاده می کنیم و در BIOS اولین وسیله راه انداز فلاپی دیسک سخت معرفی شده است، سیستم هسته موجود در دیسک سخت را بدون ارائه اعلان به کاربر در زمان راه اندازی سیستم، بارگیری خواهد کرد.

پیامهای خطای LILO

در مواردی ممکن است در زمان راه اندازی سیستم بوسیله lilo، دستگاه پس از نمایش بخشی از کلمه lilo بر روی صفحه نمایش متوقف شده و رویه راه اندازی متوقف گردد. کد راه اندازی lilo حاوی دو بخش می باشد که اولین قسمت آن در سکتور راه انداز سیستم و بخش دوم آن در فایل boot/boot.b قرار می گیرد. در طول نصب lilo، معمولاً فایلی بنام boot/map ساخته می شود که توسط آن lilo اشاره گرهای لازم جهت دسترسی به سیستم عاملهای موجود و هسته های گنو/ لینوکس را شناسایی می کند.

در زمان راه اندازی سیستم این فایل نیز باید موجود بوده و در اختیار lilo گیرد. هنگامی که lilo خودش را به داخل حافظه بارگیری می کند، کلمه lilo بر روی صفحه نمایش، نشان داده می شود. هر کدام از حروف lilo قبل و یا بعد از اتمام عملیات خاصی نمایش داده می شوند. اگر lilo در یکی از مراحل راه اندازی سیستم از کار بیفتد، حروف چاپ شده در تشخیص مشکل می توانند به ما کمک کنند.

حالت اول : عدم نمایش هیچیک از حروف LILO

در این حالت هیچکدام از اجزاء lilo به داخل حافظه بارگیری نشده است که می تواند بدلیل عدم نصب آن و یا فعال نبودن پارتیشنی باشد

که سکتور راه انداز در آن قرار گرفته است .

حالت دوم : نمایش حرف L

اولین بخش راه انداز سیستم به داخل حافظه بارگیری و اجرا گردیده است، اما نتوانسته بخش دوم راه انداز را داخل حافظه بیاورد. کدهای دو رقمی خطا در این مرحله نیز می توانند نوع مشکل را مشخص کنند . این وضعیت معمولاً در صورت وجود نقص در یکی از اجزا فیزیکی سیستم و یا بروز مشکلی در BIOS رخ می دهد . (geometry mismatch)

حالت سوم : نمایش حروف LI

اولین بخش راه انداز سیستم قسمت دوم را به داخل حافظه بارگیری کرده است ، ولی نتوانسته آنرا اجرا کند . این مشکل می تواند بدلیل جابجا کردن فایل boot/boot.b بدون اجرای نرم افزار نصب کننده فایل نقشه و یا عدم شناسایی صحیح دیسک توسط BIOS بروز کند .

حالت چهارم : نمایش حروف LIL

دومین بخش از راه انداز عملیات خود را شروع کرده است اما قادر به بارگیری جدول اشاره گر از فایل نقشه نمی باشد . این مشکل نیز عموماً بدلیل بروز مسائل فیزیکی در اجزا سیستم و یا وجود مشکلی در BIOS ایجاد می شود .

حالت پنجم : نمایش LIL?

دومین بخش از راه انداز lilo در آدرس نادرستی از حافظه بارگیری شده است . این مشکل نیز همانند حالت سوم می تواند در صورت عدم اجرای نرم افزار نصب کننده فایل نقشه پس از جابجا کردن فایل boot/boot.b صورت پذیرد . در مواردی هم BIOS می تواند موجب بروز این مشکل گردد .

حالت ششم : نمایش LIL -

جدول اشاره گر سیستم از بین رفته که دلیل بروز آن می تواند همانند حالت قبل باشد .

حالت هفتم : نمایش کلمه LILO

همه اجزا تشکیل دهنده lilo با موفقیت به داخل حافظه بارگیری و اجرا شده اند .

اغلب خطاهایی که در مراحل فوق در ارتباط با BIOS بروز می کنند، بدلیل بروز مواردی چون مشکلات فیزیکی یا وجود جداول معیوب برای پارتیشن های دیسک سیستم نمی باشند ، بلکه مشکلاتی هستند که در طول نصب lilo ایجاد شده و اغلب بدلیل در نظر نگرفتن محدوده آدرس دهی سیلندرهای دیسک سخت توسط BIOS می باشند که ۱۰۲۴ سیلندر است . ساده ترین راه حل جهت حل چنین مشکلاتی فعال سازی مد آدرس دهی گسترده دیسک یا LBA در setup سیستم است .

راه حل های موجود

در بیشتر موارد ، راه حل بکار بردن یکی از راهکارهای زیر می باشد :

۱- نصب مجدد lilo با استفاده از `/sbin/lilo` که امکان مشاهده log فایل‌های مربوط به آن در صورت نیاز وجود دارد . جهت ایجاد log فایل‌های مناسب در خروجی سیستم می توانیم از فرمان `lilo` به شکل زیر استفاده کنیم :

```
#/bin/lilo -v -v -v >/boot/lilo.log2>/boot/lilo.logerr
```

در اکثر موارد فایل `lilo.logerr` حاوی چیزی نمی باشد (به شرط صحیح بودن پیکربندی رویه راه انداز سیستم). این فایل در صورت لزوم قادر خواهد بود دقیقاً برای شما مشخص کند که `lilo` در طول راه اندازی سیستم از کدامیک از اجزا موجود در BIOS استفاده می کند .

۲- فعالسازی مد LBA در `setup` و یا نصب `lilo` در محدوده ای قبل از سیلندر ۱۰۲۴ دیسک نیز از جمله راه حل‌های احتمالی موجود می باشند . در چنین مواردی نصب مجدد هسته یا هسته های موجود بر اساس فایل‌های موجود در `/boot` و سکتور راه اندازی که حاوی `lilo` است نیز لازم است .

۳- مرور اطلاعات مربوط به پایداری دیسک سخت نیز از دیگر مواردی می باشد که به سایرین در چنین مواردی توصیه می شود . این اطلاعات را معمولاً در چهار مرحله می توان تهیه کرد .

الف - ساختار بکار رفته توسط `lilo` که با مشاهده `log` فایل‌های تشریح شده در مرحله قبل، قابل بررسی هستند

ب - ساختار تشکیل داده شده توسط هسته سیستم عامل که با مرور فایل `/var/log/messages` و یا خروجی فرمان `dmesg` می توان آنرا مرور کرد .

ج - ساختار موجود در جدول پارتیشن دیسک که توسط ابزار `fdisk` و یا فرمان `fdisk -l` قابل مشاهده است

د - ساختار بکار رفته در BIOS که `lilo` با این ساختار قبل از اینکه هر سیستم عاملی شروع به راه اندازی کند، مواجه شده و تابع آن می باشد .

در مواردی که قصد دارید تا درخواست هر گونه پشتیبانی را برای حل مشکلات خود در `lilo` داشته باشید، توصیه می شود موارد زیر را به عنوان اطلاعات اولیه در اختیار پشتیبان خود و یا هر کسی که قصد گرفتن راهنمایی از آن جهت حل مشکل را دارید، قرار دهید :

۱- فایل `/etc/lilo.conf`

۲- خروجی فرمان `fdisk -l`

۳- log فایل‌های `lilo` بر اساس آنچه در بالا گفته شد

```

dawood@linux:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

linux:/home/dawood # fdisk -l

Disk /dev/hda: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1             3076         9964     55335892+   f   W95 Ext'd (LBA)
/dev/hda2              366          487       979965    82   Linux swap / Solaris
/dev/hda3             488         1762     10241437+    83   Linux
/dev/hda4            1763         3075     10546672+    83   Linux
/dev/hda5 *          3076         3728      5245191    83   Linux
/dev/hda6            3729         4380     5237158+    83   Linux
/dev/hda7            4381         4963      4680403    83   Linux
/dev/hda8            5687         6991     10482381    83   Linux
/dev/hda9            6992         9964     23880591    83   Linux
/dev/hda10           4964         5686      5807466    83   Linux

Partition table entries are not in disk order
linux:/home/dawood #

```

خروجی فرمان fdisk

پرسشهای متداول LILO

علی رغم اینکه مستندات موجود در فایل های مرجع LILO به صورت پیش فرض در مسیر `/usr/share/doc/packages/lilo` نصب می شوند ، بسیار جامع می باشند ولی به علت پیچیدگی بیشتر کاربران نمی توانند به تنهایی فایل `/etc/lilo.conf` را خود تغییر دهند. در انتهای بحث LILO ، به بررسی چند مثال کاربردی و بررسی برخی مسائل ، که اکثر استفاده کنندگان LILO با آن مواجه هستند می پردازیم .

اطلاعات پایه و نصب استاندارد

زمانی که LILO سیستم را راه اندازی می کند ، از BIOS جهت بارگیری هسته گنو/لینوکس از روی دیسک استفاده می کند. به همین دلیل ، هسته باید در فضایی از دیسک که در دسترس BIOS می باشد ، ذخیره گردد.

زمان راه اندازی ، LILO قابلیت خواندن اطلاعات سیستم فایلها را ندارد و تنها کلیه مسیرهای تعریف شده در فایل `/etc/lilo.conf` در زمان نصب مورد استفاده قرار می گیرند. (در صورتی که بخواهید از فایل `/sbin/lilo` به عنوان فایل اجرایی lilo استفاده کنید). در زمان نصب ، برنامه lilo جدول سکتورهای مورد استفاده فایل های راه انداز سیستم عامل را بارگذاری می کند.

به طور معمول ، کلیه فایل های مذکور باید در پارتیشنی قرار بگیرند که در دسترس BIOS می باشد. مسیر پیش فرض فایل ها در `/boot` است. در نتیجه تنها پوشه مورد نیاز BIOS ، پوشه `boot` می باشد.

یادآوری : هر بار که ساختار فایل پیکربندی LILO را تغییر می دهید ، جهت اعمال تغییرات باید راه انداز سیستم را مجدداً نصب کنید. برای اینکار برنامه `/sbin/lilo` مجدداً باید اجرا گردد . همچنین هر بار که هسته سیستم عامل را مجدداً کامپایل کنید و یا تصویر کرنل را باز نویسی نمائید ، نیز نیاز به نصب مجدد LILO دارید.

در کجا باید LILO را نصب کرد؟

در فایل `/etc/lilo.conf` متغیر `boot` مسیر نصب راه انداز اصلی سیستم را مشخص می کند. در حالت کلی می توان مسیر `/dev/hda` را به عنوان پیش فرض به متغیر `boot` نسبت داد و یا از مسیر پارتیشن `root` که معمولاً `/dev/hda1` و یا `dev/hda2` استفاده کرد . در صورتی که سیستم عامل دیگری را روی سیستم خود نصب کرده باشید ، بهتر است LILO را در پارتیشن `root` نصب کنید و حتی الامکان در MBR نصب نکنید.

در این صورت ، پارتیشنی که اطلاعات مذکور را در خود دارد به عنوان پارتیشن قابل راه اندازی سیستم و یا "bootable" نامیده می شود که در دستور `fdisk` با استفاده از پارامتر "a" و در دستور `cfdisk` با استفاده از پارامتر "b" در دستور می توان پارتیشن ها را bootable نمود.

با فشردن حرف b از صفحه کلید ، عبارت boot بر روی پارتیشن مورد نظر ظاهر خواهد شد

مسلم است که همواره راههای میانبری برای اجتناب از دستورات فوق وجود دارد.

شما می توانید LILO را روی MBR نصب کنید. حتی اگر سیستم عامل دیگری را روی دستگاه خود داشته باشید.

به عنوان مثال ، اگر شما Windows را به عنوان اولین سیستم عامل روی ماشین خود نصب کرده باشید ، راه انداز Windows روی MBR

اطلاعات لازم جهت راه اندازی سیستم Windows را قرار می دهد. حال فرض کنید که پس از نصب Windows ، سیستم عامل Linux را

نصب کرده باشید و بخواهید که LILO در MBR نصب شود، در این صورت راه انداز سیستم Windows حذف شده و راه انداز سیستم عامل گنو/لینوکس جایگزین آن می شود. پس از نصب گنو/لینوکس و بازنویسی MBR در صورتی که مجدداً سیستم را روشن کنید، شما قادر به راه اندازی Windows نخواهید بود. ولی این عمل مشکلی ایجاد نمی کند. تنها کافیسست فایل `/etc/lilo.conf` را تصحیح نموده و پیکربندی لازم Windows را به آن اضافه کنید. سپس با اجرای فرمان `lilo` و راه اندازی مجدد سیستم، در منوی اولیه سیستم به همراه گزینه گنو/لینوکس، گزینه Windows نیز ظاهر خواهد شد.

چگونه می توان درایورهای هارد IDE را پیکربندی کرد؟

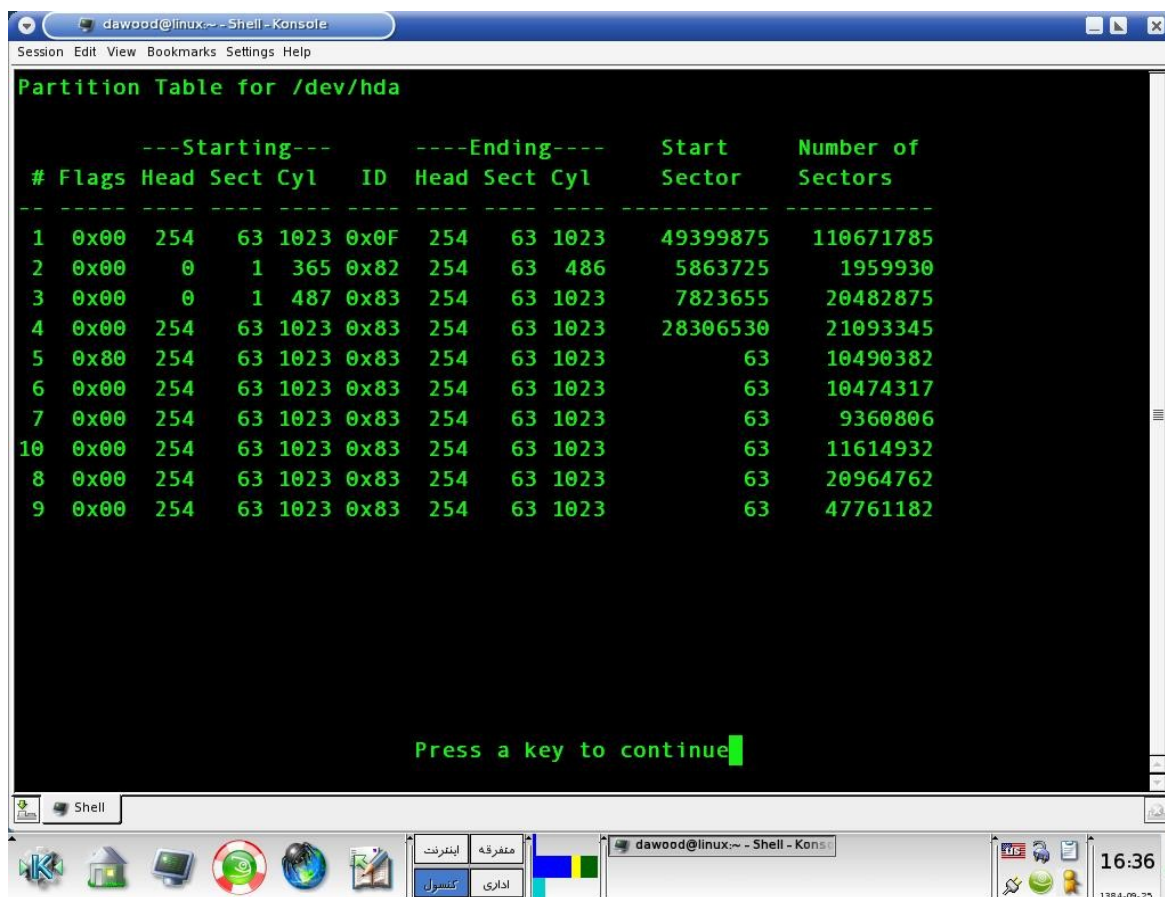
فرض کنید که مایل به استفاده از مد LBA و یا تنظیمات دیسکهای با حجم بالا در BIOS خود نیستید و تنها قصد نصب گنو/لینوکس رادرسیتهم خود دارید، در این صورت باید هسته مرکزی سیستم عامل گنو/لینوکس را در فضای ۱۰۲۴ سیلندر اول دیسک قرار دهید. این مسئله تا مادامی که درایوهای خود را پارتیشن بندی کرده و حجم `root` را پایین نگه می دارید مشکلی را برای شما ایجاد نمی کند.

PhoenixBIOS Setup Utility		
Main		
Primary Master [Primary Master]		Item Specific Help
Type:	[User]	
Cylinders:	[0]	
Heads:	[1]	
Sectors:	[0]	
Maximum Capacity:	0MB	
LBA Format		
Total Sectors:	0	
Maximum Capacity:	0MB	
Multi-Sector Transfers:	[Disabled]	
LBA Mode Control:	[Disabled]	
32 Bit I/O:	[Disabled]	
Transfer Mode:	[Standard]	
Ultra DMA Mode:	[Disabled]	
F1 Help	↑↓ Select Item	-/+ Change Values
Esc Exit	↔ Select Menu	Enter Select ► Sub-Menu
F9	Setup Defaults	
F10	Save and Exit	

فعالسازی مد LBA در BIOS های قدیمی از طریق Setup

ولی در صورتی که دیسک سخت شما در حال حاضر شامل سیستم عامل دیگری غیر از گنو/لینوکس نیز می باشد، شما قادر به تنظیم مقادیر BIOS نبوده و یا سیستم عامل قبلی شما بدون اعمال تغییرات بالا نخواهد آمد. توجه داشته باشید که کلیه توزیع های LILO قابلیت سازگاری با مد LBA و تنظیمات دیسک های با حجم بالا را دارا می باشند.

لازم به ذکر است در فایل `/etc/lilo.conf` گزینه "linear" مشکلات آدرس دهی را نیز برطرف می کند که با استفاده از آن ساختار LILO از آدرس دهی سه بعدی حاصل از مضرب مقادیر cylinder/head/sector به آدرس دهی خطی تبدیل می شود. تبدیل آدرس دهی خطی به سه بعدی در زمان اجرا انجام می شود. در نتیجه استفاده از این پارامتر linear مشکلات آدرس دهی سه بعدی را برطرف می کند.



Partition Table for /dev/hda

#	Flags	---Starting---			ID	----Ending----			Start Sector	Number of Sectors
		Head	Sect	Cyl		Head	Sect	Cyl		
1	0x00	254	63	1023	0x0F	254	63	1023	49399875	110671785
2	0x00	0	1	365	0x82	254	63	486	5863725	1959930
3	0x00	0	1	487	0x83	254	63	1023	7823655	20482875
4	0x00	254	63	1023	0x83	254	63	1023	28306530	21093345
5	0x80	254	63	1023	0x83	254	63	1023	63	10490382
6	0x00	254	63	1023	0x83	254	63	1023	63	10474317
7	0x00	254	63	1023	0x83	254	63	1023	63	9360806
10	0x00	254	63	1023	0x83	254	63	1023	63	11614932
8	0x00	254	63	1023	0x83	254	63	1023	63	20964762
9	0x00	254	63	1023	0x83	254	63	1023	63	47761182

Press a key to continue

نمایش مشخصات پارتیشن های اولین دیسک سیستم بر اساس Head/Sect/Cyl های ابتدایی و انتهایی آنها

در صورتیکه بیش از یک دیسک سخت داشته باشید و بعضی از آن ها کاملاً در اختیار سیستم عامل گنو/ لینوکس قرار گرفته باشند، ولی در روال راه اندازی نقشی را ایفا نمی کنند، می توانید برای BIOS آنها را نصب نشده معرفی کنید. در این صورت سیستم شما سریعتر راه اندازی شده و بعد از راه اندازی لینوکس به صورت خودکار کلیه دیسک ها را در زمان کوتاهی شناسایی می کند.

نحوه عملکرد در زمان راه اندازی

زمانی که علامت چشمک زن LILO را ملاحظه می کنید، با فشار کلید tab می توانید لیست گزینه های قابل انتخاب را مشاهده کنید. در صورتی که LILO جهت پاسخگویی به درخواستهای شما پیکربندی نشده باشد، قبل از ظاهر شدن پیام LILO کلید Alt و یا Shift را

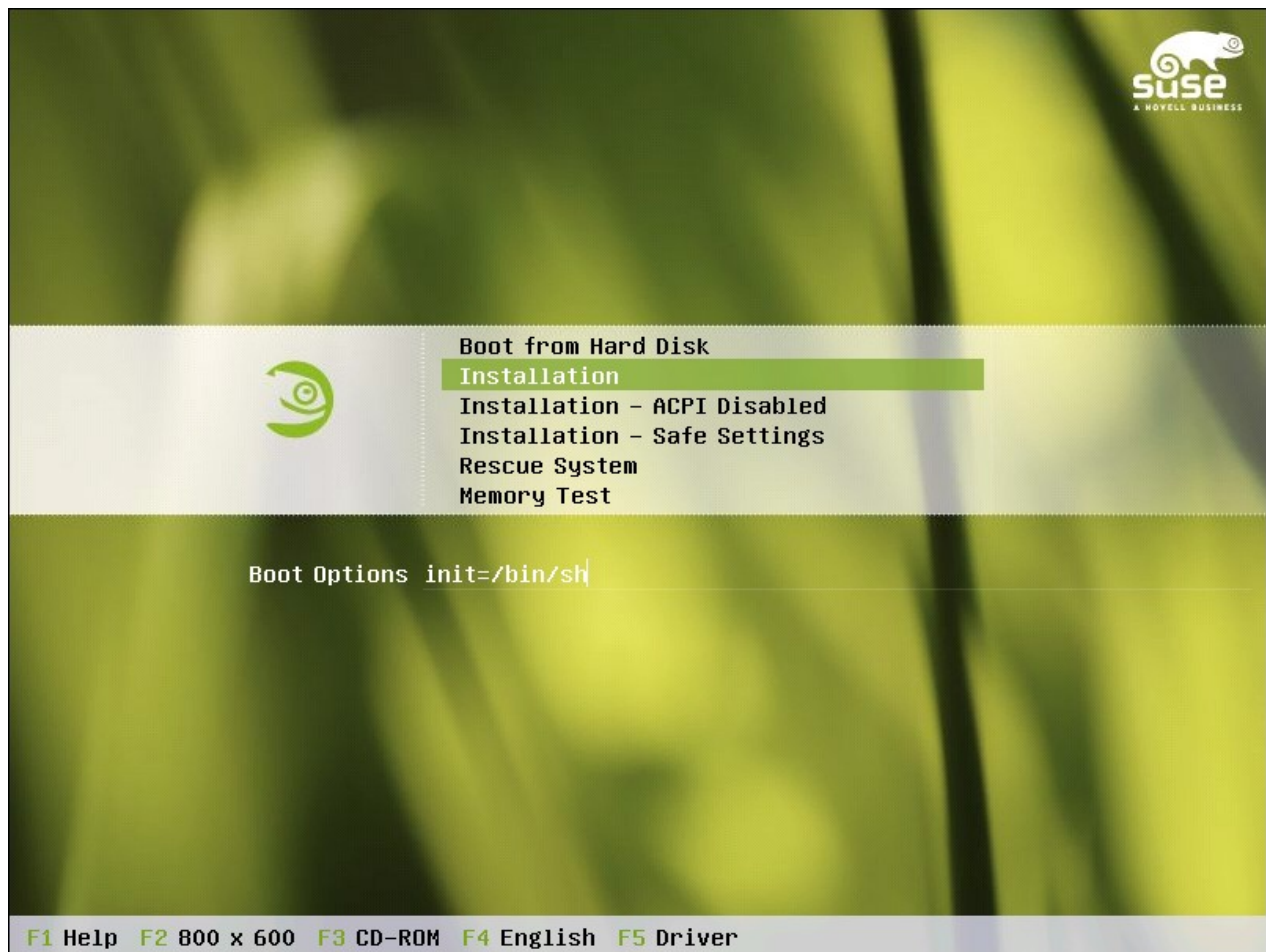
پایین نگه دارید .

همچنین هرگاه قصد راه اندازی هسته مرکزی گنو/ لینوکس دارید ، می توانید از دستورهای خط فرمان پس از نام سیستم مورد نظر خود استفاده کنید . هسته مرکزی لینوکس بسیاری از دستورهای خط فرمان را در زمان راه اندازی سیستم می پذیرد (کلیه آرگومانهای مورد استفاده در این قسمت در کتاب Boot Prompt HowTo اثر پائول گورت مکر آورده شده است) . آرگومانهای مورد استفاده و مهم در خط فرمان به شرح زیر می باشد :

root = : در زمان راه اندازی سیستم شما می توانید به هسته مرکزی سیستم عامل گنو/ لینوکس پارتیشن دیگری را به عنوان پارتیشن root معرفی کنید . پارتیشن پیش فرض جهت راه اندازی سیستم در فایل `etc/lilo.conf` / قرار دارد . به طور مثال می توانید یک پارتیشن با حداقل اطلاعات مورد نیاز جهت راه اندازی لینوکس در سیستم خود ایجاد کنید تا هرگاه که پارتیشن root اصلی خودتان در اثر اشتباه خراب شد ، شما قادر به راه اندازی سیستم از روی پارتیشن کمکی باشید .

init = : در صورتی که هنگام راه اندازی سیستم با مشکلات جدی مواجه شده باشید می توانید از دستور `init = /bin/sh` جهت حل پاره ای از مسائل استفاده کنید . این دستور معادل دستور `mount -a ; mount -w -n -o remount` در خط فرمان است . فراموش نکنید که پیش از خاموش کردن سیستم دستور `umount -a` را وارد نمایید . کاربرد اساسی دستور `init = /bin/sh` در زمان راه اندازی سیستم ، مواردی است که کلمه عبور کاربر ریشه را فراموش کرده اید و توسط آن می توانید براحتی کلمه عبور کاربر ریشه را مجدداً تغییر دهید .

اعداد : با تخصیص اعداد در خط فرمان هسته مرکزی سیستم عامل شما برنامه `init` را وارد سطوح اجرایی تعریف شده می کنید . بازه اعداد انتخابی از صفر تا شش می باشد که معادل هفت سطر اجرایی در سیستم لینوکس است . جهت کسب اطلاعات بیشتر از مستندات `init` استفاده کنید و ساختار فایل های `/etc/rc*.d` و `/etc/inittab` را مطالعه نمایید .



استفاده از فرمان **init** در زمان راه اندازی سیستم در بارگذار راه انداز توزیع **SuSE**

چگونگی Unistall کردن LILO

هنگامی که LILO اطلاعات سکتور راه انداز (boot) را بازنویسی می کند ، یک نسخه پشتیبان از فایل قبلی را در مسیر boot /boot.xxyy ذخیره می کند . اعداد xxyy شماره major و minor ، وسیله مورد نظر در مبنای شانزده می باشد . شما می توانید اعداد major و minor موجود در دیسک و یا پارتیشن دلخواه را با استفاده از دستور `ls -l /dev/hda` بدست آورید .

به عنوان مثال اولین سکتور (major 3 , minor 0) `dev/hda` تحت عنوان نام `boot/boot.0300` ذخیره می شود . نصب LILO روی `dev/fd0` باعث ایجاد فایل `boot/boot.0200` می شود و نصب LILO در (major3,minor19) `dev/sdb3` باعث ایجاد فایل `boot/boot.0319` می گردد . لازم به ذکر است که شما خود فایل مذکور را ایجاد نمی کنید ، لذا نیازی به انجام عملیات اضافی جهت تهیه پشتیبان هنگام نصب مجدد LILO ندارید .

```
nahaja@XANEWAMEQ31: /boot - Shell - Console
Session Edit View Bookmarks Settings Help

Linux: /# cd /boot/
Linux: /boot# ls -la
total 2388
drwxr-xr-x  2 root root    4096 2005-12-17 12:53 .
drwxr-xr-x 23 root root    4096 2005-08-23 03:18 ..
-rw-r--r--  1 root root     512 2005-08-23 03:16 boot.0300
-rw-r--r--  1 root root     512 2005-08-23 03:16 boot.0306
-rw-r--r--  1 root root   50355 2004-11-29 10:33 config-2.6.9-x1
-rw-r--r--  1 root root  151524 2005-08-23 03:16 initrd-2.6.9-x1.gz
-rw-r--r--  1 root root  115712 2005-12-01 09:46 map
lrwxrwxrwx  1 root root     16 2005-08-23 03:04 splash.lilo -> xandros.splash.b
lrwxrwxrwx  1 root root     19 2005-08-23 03:11 System.map -> System.map-2.6.9-x1
-rw-r--r--  1 root root  822283 2004-11-29 10:33 System.map-2.6.9-x1
-rw-r--r--  1 root root 1236702 2004-11-29 10:33 vmlinuz-2.6.9-x1
-rw-r--r--  1 root root   24434 2004-12-04 17:41 xandros.splash.b
Linux: /boot#
```

محتویات شاخه /boot در سیستم دارای بارگذار راه انداز LILO

```
dawood@linux:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

dawood@linux:~> ls -l /dev/hda
brw-r----- 1 root disk 3, 0 2005-12-15 19:08 /dev/hda
dawood@linux:~> ls -l /dev/hda1
brw-r----- 1 root disk 3, 1 2005-12-15 19:08 /dev/hda1
dawood@linux:~> ls -l /dev/hda2
brw-r----- 1 root disk 3, 2 2005-12-15 19:08 /dev/hda2
dawood@linux:~> ls -l /dev/hda5
brw-r----- 1 root disk 3, 5 2005-12-15 19:08 /dev/hda5
dawood@linux:~> ls -l /dev/hdc
brw-rw----+ 1 dawood disk 22, 0 2005-12-15 19:08 /dev/hdc
dawood@linux:~> ls -l /dev/sda
brw-rw----+ 1 root disk 8, 0 2005-12-16 15:29 /dev/sda
dawood@linux:~> ls -l /dev/sda1
/bin/ls: /dev/sda1: No such file or directory
dawood@linux:~> ls -l /dev/hdc2
/bin/ls: /dev/hdc2: No such file or directory
dawood@linux:~>
```

بدست آوردن

مقادیر major و minor با استفاده از مشاهده خصوصیات دیسک و پارتیشن‌ها

کلیه کپی های موجود در boot / تصویر آخرین وضعیت سیستم قبل از تغییر LILO می باشد . در صورتی که نیازمند uninstall کردن LILO هستید ، تنها کافی است سکتور boot را بازبایی کنید . به طور مثال اگر LILO رادرمسیر /dev/hda نصب کرده اید کافی است دستور زیر را اجرا نمایید :

```
dd if = /boot/boot.0300 of = /dev/hda bs = 446 count = 1
```

و یا اینکه از دستور زیر استفاده کنید :

```
cat /boot/boot.0300 > /dev/hda
```

همچنین با از استفاده سوئیچ -u فرمان lilo ، می توان بارگذار راه انداز LILO را از MBR سیستم حذف کرد . به طور مثال برای حذف LILO از اولین دیسک سخت سیستم داریم :

```
# lilo -u /dev/hda
```

این دستورات ساده تر از دستور **fdisk /mbr** در محیط DOS می باشند زیرا به شما اجازه می دهد به سادگی لینوکس را از روی دیسک حذف کنید بدون آنکه نیاز به راه اندازی مجدد سیستم داشته باشید .

پیکربندی ساده lilo.conf

در اکثر فایل های lilo.conf از پیکربندی زیر برای بارگذار راه انداز LILO استفاده می شود :

```
boot = /dev/hda      # or your root partition

delay = 10           # delay , in tenth of a second

vga =0               # optional

# linear             # try "linear" in case of geometry problems

image = /boot/vmlinux  #your zimage file

root = /dev/hda1      # your root partition

label =linux          # or any fancy name

read-only             # mount root read-only
```

```
other = /dev/hda4      # your dos partition ; if any

table = /dev/hda       # the current partition table

label = dos            # or any non-fancy name
```

در صورت نیاز می توانید چندین image و بخشهای مختلف برای سیستم عامل های مورد استفاده خود در سیستم تعریف کنید . البته داشتن چندین تصویر مختلف از هسته مرکزی سیستم عامل با نگارشهای مختلف ، امری چندان متداول نیست . (در صورتی که هسته مرکزی سیستم عامل همواره به روز باشد)

روش نصب Windows از منوی LILO

در این قسمت روش نصب windows و لینوکس را تحت سیستمی با راه انداز lilo شرح می دهیم .

در آغاز ، یک نسخه ی کامل از سیستم عامل windows را روی سیستم نصب کنید. بدیهی است که قبل از نصب ، از کلیه اطلاعات مهم خود پشتیبان تهیه کنید تا در موقع نصب windows دچار مشکل نشوید.

در هنگام نصب windows ، به صورت پیش فرض مکان نصب راه انداز سیستم از شما پرسیده نمی شود و اطلاعات در MBR ذخیره می گردد. ولی امکان باقی ماندن نگارش های قبلی MBR به همان شکل پیشین آن نیز وجود دارد. در نتیجه پیشنهاد می شود که سیستم را با فلاپی راه انداز DOS که حاوی ابزار لازم برای استفاده از فرمان fdisk است ، راه اندازی کرده و پس از ملاحظه علامت A:\ دستور fdisk /mbr را وارد نموده و سیستم را این بار بدون فلاپی روشن کنید.

پس از نصب موفقیت آمیز windows ، ملاحظه می کنید که قسمتی از فضای دیسک و یا تمام فضای دیسک شما به این سیستم عامل اختصاص یافته است . توصیه می شود که برای نصب گنو/ لینوکس از فضای خالی دیسک استفاده کنید و حجم ویندوز را مجدداً تغییر ندهید. ولی در صورت نیاز به تغییر فضای ویندوز به روش زیر عمل کنید:

پس از پیکربندی windows و راه اندازی آن سیستم را با فلاپی و یا CD حاوی نرم افزار partition magic و یا power quest راه اندازی کنید. این نرم افزارها ابزاری گرافیکی ، جهت نمایش اطلاعات کلیه فضای روی دیسک را به شما می دهند . بهترین قابلیت این نرم افزارها فراهم آوردن امکان تغییر حجم پارتیشنهای سیستم بدون از بین رفتن اطلاعات آنها است. از جمله دیگر ابزار های قابل استفاده برای این کار ، نرم افزار FIPS می باشد که با وجود حجم پایین خود این قابلیت را داراست.

یکی از راههای موجود کاهش حجم پارتیشنی خاص از سیستم ، جهت فراهم آوردن فضای خالی است. علی رغم کلیه امکانات و قابلیت های نرم افزارهای مذکور توصیه می شود که قبل از تغییر فضای پارتیشن ها از اطلاعات خود نسخه پشتیبان تهیه کنید.

معمولاً اگر بلافاصله پس از نصب windows اقدام به تغییر فضای پارتیشن ها کنید ، به علت اینکه هنوز اطلاعات خاصی روی فضای دیسک ذخیره نشده است ، در صورت بهم ریختگی سیستم تنها کافی است مجدداً ویندوز را نصب کنید و دیگر نیازی به بازیابی اطلاعات ندارید.

به این ترتیب نرم افزار partition magic و یا هر یک از نرم افزارهای مشابه که با آنها آشنایی دارید ، می تواند فضای سیستم عامل و پارتیشن های شما را تغییر دهد . بطور مثال اگر سیستم عامل شما Windows است ، partition magic قادر است دیسک سخت شما را با سیستم فایل NTFS و یا FAT تغییر حجم دهد که این تغییر حجم را می توان از ابتدا و یا انتهای کل فضای اشغال شده توسط Windows اعمال کرد . بدین معنا که می توانید فضای آزاد شده را قبل از فضای اشغال شده سیستم در نظر بگیرید و یا اینکه پس از فضای اشغال شده توسط سیستم عامل فضای آزاد خود را تعریف کنید. معمولاً فضای آزاد را در قسمت آخر دیسک تعریف می کنند.

پس از اتمام کار سیستم عامل Windows خود را با نرم افزارهایی مثل windows explorer و یا disk administrator بررسی کنید تا از صحت اعمال تغییرات اطمینان پیدا کنید.

حال قدم بعدی را جهت نصب لینوکس برمی داریم . برای نصب کافی است CD راه انداز که اولین دیسک می باشد را در CD-ROM قرار داده و سیستم را روشن کنید تا نصب آغاز شود . حال می توانید گزینه های مختلف نصب را انتخاب کنید. مثلاً نوع نرم افزارهای مورد نظر، نحوه پیکربندی سیستم و ...

لازم به ذکر است ، کلیه ساختارهایی که اکنون انتخاب می کنید، مانند نام نرم افزارهای نصب شده و ... پس از نصب قابل تغییر است. جهت نصب راه انداز سیستم گزینه LILO را انتخاب کنید. پس از نصب ، LILO در MBR قرار می گیرد. حال سیستم خود را مجدداً راه اندازی کنید. در آغاز راه اندازی سیستم ، شما ملاحظه می کنید که منوی LILO با نمایش یک سیستم عامل لینوکس که هم اکنون نصب شده ظاهر می شود. نگران نباشید. سیستم عامل ویندوز شما پاک نشده است . تنها کافیست فایل lilo.conf را که در مسیر `/etc/lilo.conf` قرار دارد ویرایش کنیم.

زمانی که این فایل را باز می کنید ملاحظه می کنید که فقط یک موجودیت برای شناسایی لینوکس تعریف شده است. (در صورتی که چند سیستم عامل گنو/ لینوکس دیگر هم روی سیستم از قبل نصب کرده باشید، مشخصات آن سیستم عامل ها را نیز در این فایل خواهید یافت) حال کافی است محل دقیق سیستم عامل ویندوز خود را به همراه پارامترهای لازم به فایل `/etc/lilo.conf` اضافه کرده و سپس فرمان `lilo` را اجرا نمایید . پس از انجام تغییرات و ذخیره آنها سیستم عامل را مجدداً راه اندازی کرده و در منوی اولیه LILO سیستم

عامل ویندوز و گنو/لینوکس را ملاحظه کنید. درج این نکته نیز لازم است که در اکثر توزیع هایی که امروزه در آنها از LILO به عنوان بارگذار راه انداز سیستم استفاده می شود (مانند توزیع Xandros) ، معمولاً پس از نصب توزیع سایر سیستم عامل های موجود در سیستم به شکل خودکار شناسایی شده و در فایل پیکربندی lilo و منوی راه اندازی سیستم قرار می گیرند. در چنین مواردی نیازی به ویرایش فایل lilo.conf نخواهیم داشت .

استفاده از LILO زمانی که BIOS قادر به دیدن پارتیشن ریشه نمی باشد

فرض کنید دو دیسک IDE و یک دیسک SCSI توسط BIOS دیده نمی شود . راه انداز گنو/لینوکس ، یعنی LILO با استفاده از فراخوانهای BIOS می تواند تنها دیسک هایی را ببیند که BIOS قادر به دیدن آنهاست . فرض کنید در حالی که BIOS تنها دو دیسک IDE را می بیند، راه انداز اصلی و سیستم فایل root بر روی دیسک SCSI نصب شده است. راه حل این مشکل ذخیره سازی هسته مرکزی سیستم عامل ، فایل های نقشه و یک زنجیره راه انداز برای پارتیشن گنو/لینوکس روی اولین دیسک IDE است . لازم به ذکر است که نیازی به نگهداری از هسته مرکزی سیستم عامل بر روی پارتیشن root نمی باشد . اگر دومین پارتیشن بر روی اولین دیسک IDE قرار داشته باشد و بر روی شاخه /u2 ، mount شده باشد، فایل پیکربندی lilo.conf بصورت زیر خواهد بود .

```
boot = /dev/hda
```

```
install = /u2/etc/lilo/boot.b
```

```
message = /u2/etc/lilo/message
```

```
map = /u2/etc/lilo/map
```

```
compact
```

```
prompt
```

```
timeout = 100
```

```
image = /u2/z1.2.1
```

```
label = 1.2.1
```

```
root = /dev/sda1
```

```
read-only
```

image = /u2/z1.0.9

label = 1.0.9

root = /dev/sda1

read-only

other = /dev/hda1

loader = /u2/etc/lilo/chain.b

label = windows

alias = m

مرور دقیق تر بازیابی یک سیستم آسیب دیده توسط یک دیسک زنده

مواردی ممکن است رخ دهد که به دلیل اشتباهات سهوی و یا استفاده نادرست از ابزارها ی پیکربندی سیستم، MBR سیستم آسیب دیده و شما قادر به راه اندازی سیستم عاملهای خود نباشید . در چنین مواردی معمولاً در زمان راه اندازی تنها پیغامی بر روی صفحه نمایش مشاهده می کنید که بر عدم وجود برنامه راه انداز سیستم اشاره می کند و یا از شما وسیله دیگری جهت راه اندازی سیستم درخواست می نماید .

از جمله راه حل هایی که جهت حل چنین مشکلی در سایر توزیع های گنو/لینوکس ارائه می گردد ، استفاده از دیسکهای زنده (Live CD) و راه اندازی سیستم از روی آنها می باشد . با راه اندازی سیستم توسط یک دیسک زنده قادر خواهیم بود با قراردادن یک سیستم عامل در حافظه رایانه به تمام منابع و فایل های موجود در دیسک سخت سیستم خود دسترسی پیدا کنیم . از جمله توزیع هایی که از آنها عموماً برای اشکال زدایی سیستم های آسیب دیده و رفع مشکلاتی این چنینی می توان استفاده کرد، توزیع های زنده Ubuntu,Knoppix,Slax می باشند . البته از هر توزیع کننده دیگری که دارای یک برنامه راه انداز سیستم، جهت نصب مجدد در سیستم آسیب دیده می باشد نیز می توان جهت بازسازی سیستم استفاده کرد حال بصورت کلاسیک به بررسی رویه بازیابی سیستم آسیب دیده و نصب یک راه انداز GRUB در سیستمی که حاوی چندین سیستم عامل مختلف می باشد می پردازیم :

۱- ابتدا دیسک زنده مورد نظر خود را در CD-ROM ، خود قرارداده و سیستم را مجدداً راه اندازی می کنیم . (دقت داشته باشید که در

setup سیستم ، اولین وسیله راه اندازی سیستم را CD-ROM تعریف کرده باشید)

۲- با راه اندازی سیستم از روی CD-ROM ، سیستم عامل موجود در دیسک زنده به داخل حافظه RAM رایانه بارگیری شده و پس از طی مراحل اولیه راه اندازی وارد سیستم عامل دیسک زنده می شوید که و از آن می توانید امکانات لازم جهت دستیابی به منابع سیستم را تهیه کنید .

۳- با بازکردن یک کنسول و یا وارد شدن به یکی از ترمینالهای موجود (tty) قادر خواهیم فرامین مورد نظر را جهت بازسازی راه انداز سیستم اعمال نماییم .

* نکته : در صورتی که در حال اجرای این روند در سیستمی هستید که اطلاعات چندانی از دیسک آن ندارید بهتر است قبل از انجام هر کاری توسط فرمان fdisk -l اطلاعاتی راجع به پارتیشنهای آن کسب کنید . خروجی این فرمان شامل پارتیشن های موجود در دیسک سخت به همراه نوع آنها ، سیلندرهای ابتدایی و انتهایی آنها و اندازه آنها بر حسب بایت می باشد.

۴- حال با اجرای فرمان grub تحت کاربری به ابزار لازم جهت بازسازی MBR دست پیدا کرده و تنظیمات لازم را باید در آن اعمال نماییم. دربرخی از توزیع ها مانند RedHat ، اگر در زمان راه اندازی سیستم با مشکل مواجه شویم ، این ابزار که همان پوسته grub می باشد به شکل خودکار در اختیار ما قرار می گیرد.

۵- پس از گرفتن اعلان grub> باید فرمان زیر را تایپ کرده و خروجی آن را مورد بررسی قرار دهیم:

```
grub> find /boot/grub/stage1
```

خروجی این فرمان می تواند چندین پارامتر به صورت (hdx,y) باشد که همانطور که قبلاً اشاره شده X,Y اعدادی صحیح و مثبت بوده و مقدار X علاوه یک شماره دیسک و مقدار Y علاوه یک شماره پارتیشنی از سیستم که فایل اولیه راه انداز GRUB در آن قرار دارد را نشان می دهد. حال از بین خروجی های موجود ، باید پارتیشنی که می خواهیم براساس تنظیمات موجود در فایل menu.lst آن ، MBR سیستم مجدداً بازسازی گردد را انتخاب نموده و در صورتی که راه انداز سیستم نصب نماییم.

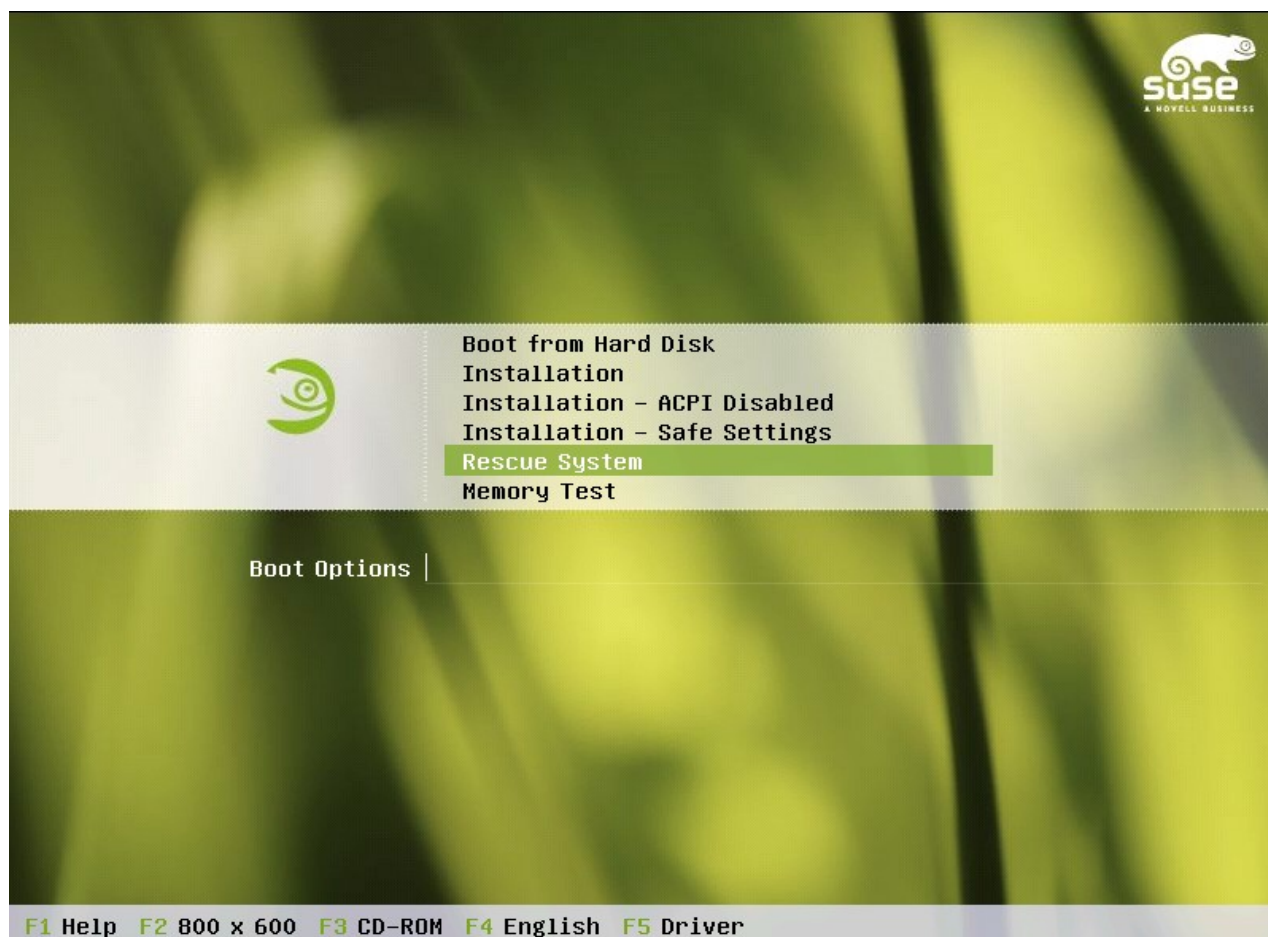
نکته : جهت اطمینان از انتخاب پارتیشن مورد نظر و فایل menu.lst می توانید محتویات فایل مذکور را در سایر پارتیشن های سیستم مشاهده نموده و پس از اطمینان از صحت انتخاب مراحل زیر را دنبال کنیم . در مواردی که توسط مُد Rescue اولین دیسک برخی از توزیع ها مانند RedHat یا SuSE اقدام به بازسازی MBR می نماییم ، جهت مشاهده محتویات سایر پارتیشن ها ابتدا باید آنها را توسط فرمان mount به سیستم فایل موجود در حافظه متصل نموده و سپس در صورت لزوم توسط فرمان chroot پارتیشن مذکور را به شرط ریشه بودن به عنوان سیستم فایل ریشه جاری خود تعریف نماییم.

انتخاب مد Rescue در توزیع SuSE هنگام راه اندازی سیستم توسط اولین CD

مثالی از این فرآیند را در شکل زیر می توانید مشاهده کنید:

```
#mount /dev/hda2 /mnt
```

```
#chroot /mnt
```



با اجرای فرمان `chroot` پارتیشن ریشه ما از سیستم فایل موجود در حافظه که یک سیستم فایل مجازی می باشد، و از روی اولین CD یک توزیع گنو/ لینوکس ایجاد شده است به `/dev/hda2` تغییر پیدا کرده و خروجی فرمان `ls /dev/hda1` را به ما نشان می دهد.

۶) با مشخص شدن پارتیشن موردنظر توسط فرمان `root` به شکل زیر می بایست آن را به سیستم معرفی کرده و سپس توسط فرمان `setup` در MBR دیسک ، راه انداز سیستم را مجدداً نصب می کنیم. دستور `setup` در پشت صحنه از فرمان قدرتمند `install` استفاده می کند که

درنهایت منجر به بازیابی سیستم آسیب دیده می گردد.

```
grub> root (hd0,2)
```

```
grub> setup (hd0)
```

با انجام مراحل فوق و اجرای فرمان quit از پوسته GRUB خارج شده و با راه اندازی مجدد سیستم توسط فرمان reboot قادر خواهیم بود MBR سیستم را مجدداً راه اندازی کنیم. البته به یاد داشته باشید که پس از راه اندازی سیستم ، دیسک زنده را از CDROM خارج نمایید تا سیستم مجدداً از طریق آن راه اندازی نشود. !

مراجع

- <http://portal.suse.com/>
- <http://www.wikipedia.org>
- <http://www.gnu.org/software/grub>
- <http://www.linuxjournal.com/article>
- <http://www.redhat.com/docs/manuals/linux>
- <http://www.ibiblio.org/pub/linux/docs/Howto/other-formats/html-single/partition.html>
- http://www.gentoo-wiki.com/HOWTO_Dual-boot_when_installing_Windows_after_Gentoo
- SuSE Professional 9.3
& 10 Admin guide , Novell Inc (Edition 3 – 2005)
- Linux System
Administration (Second Edition) Vicki stanfield , Roderick W .Smith