

Linux Professional Institute

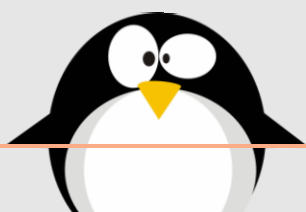
LPIC-1

جزوه‌ی پنجم: دستگاه‌های ذخیره‌سازی و
پارتیشن‌بندی آنها

By: The Albatross

thealbatross@yandex.com

<https://github.com/TheAlbatrossCodes/Linux-In-Persian>



فهرست مطالب

مقدمه.....	۱
ذخیره سازی داده	۱
طریقه ی اتصال درایوها به لینوکس	۱
پارتیشن بندی درایوها.....	۲
سیستم های پارتیشن بندی.....	۳
MBR.....	۳
GPT.....	۵
تشخیص اتوماتیک درایوها.....	۶
Mount Point ها.....	۷
پارتیشن بندی هنگام نصب سیستم	۹
مشاهده ی اطلاعات پارتیشن بندی.....	۱۶
مشاهده ی اطلاعات پارتیشن بندی با دستور <i>lsblk</i>	۱۶
مشاهده ی اطلاعات پارتیشن بندی با استفاده از <i>fdisk</i>	۱۷
مشاهده ی اطلاعات پارتیشن با استفاده از <i>parted</i>	۱۸
تحقیق: LVM چیست؟	۱۸

مقدمه

جلسه‌ی قبل به طور کامل در مورد پراسس‌ها و چگونگی مدیریت آنها صحبت کردیم و سپس توضیحاتی در مورد سخت‌افزارهای کامپیوتر و چگونگی مدیریت آنها دادیم. در این جلسه در مورد چگونگی برخورد لینوکس با سخت‌افزارهای مخصوص ذخیره‌سازی داده (هارددرایوها و...) صحبت می‌کنیم و سپس در مورد سیستم‌های متفات پارتیشن‌بندی و چگونگی پارتیشن‌بندی هنگام نصب یک سیستم لینوکس صحبت می‌کنیم.

ذخیره‌سازی داده

معمول‌ترین روش ذخیره‌ی اطلاعات بر روی سیستم، استفاده از یک Hard Disk Drive یا HDD می‌باشد. HDDها دستگاه‌های فیزیکی هستند که اطلاعات را به صورت مغناطیسی بر روی یک سری صفحه چرخنده ذخیره می‌کنند. امروزه روشی دیگر برای ذخیره‌ی اطلاعات نیز وجود دارد که Solid-State Drive یا SSD نام دارد. SSDها از مدارهای الکترونیکی جهت ذخیره‌ی اطلاعات استفاده می‌کنند. برخلاف HDDها، در SSD هیچ قطعه‌ی متحرکی وجود ندارد. این امر باعث می‌شود که SSDها سریع‌تر و قابل اعتمادتر باشند. در حال حاضر SSDها گران‌تر از HDDها می‌باشند، ولی این امر به سرعت در حال تغییر است (البته در ایران هر دو به موازات هم گران‌تر می‌شوند) و پیش‌بینی می‌شود که HDDها به زودی به صورت کامل منسوخ شوند. از نظر سیستم‌عامل لینوکس، HDD تفاوتی با SSD ندارد و رفتار لینوکس با آنها هیچ تفاوتی ندارد. نکته‌ای که برای لینوکس مهم است، طریقه‌ی اتصال این دستگاه‌ها به سیستم می‌باشد.

طریقه‌ی اتصال درایوها به لینوکس

با این که HDDها و SSDها در طریقه‌ی ذخیره‌ی داده با هم تفاوت دارند، اما هر دو با روش‌های مشابهی به لینوکس متصل می‌شوند. به طور کلی، سه روش برای اتصال درایوها به لینوکس وجود دارد:

- Parallel Advanced Technology Attachment یا PATA که درایوها را از طریق اینترفیس موازی به سیستم متصل می‌کند. احتمالاً این نحوه‌ی اتصال را در کامپیوترهای قدیمی‌تر دیده باشید؛ این نوع اتصال به یک کابل پهن نیاز دارد.
- Serial Advanced Technology Attachment یا SATA که درایوها را از طریق اینترفیس سریال به سیستم متصل می‌کند. سرعت SATA بسیار بالاتر از PATA می‌باشد.
- Small Computer System Interface یا SCSI که درایوها را از طریق اینترفیس موازی به سیستم متصل می‌کند، اما سرعتش مانند SATA می‌باشد.

وقتی که یک درایو را به لینوکس متصل می‌کنیم، کرنل لینوکس به آن درایو یک فایل در دایرکتوری `/dev` اختصاص می‌دهد. به هر کدام از این فایل‌ها، Device File می‌گویند. هر داده‌ای که روی دیوایس فایل نوشته شود، مستقیماً درون درایو قرار می‌گیرد و همچنین اگر آن فایل را بخوانیم، دقیقاً اطلاعات موجود روی دستگاه را خوانده ایم.

برای دستگاه‌هایی که از طریق PATA به سیستم متصل شده باشند، نام دیوایس فایل به صورت `/dev/hdx` خواهد بود. به طوری که `x` نشان دهنده‌ی هر دستگاه یا درایو منحصر به فرد می‌باشد. برای مثال اولین درایو

متصل شده به سیستم از طریق PATA، hda نام میگیرد، دومین درایو PATA، hdb نام میگیرد و نام گذاری دستگاه‌های PATA به همین ترتیب ادامه می‌یابد.

نام دیوایس فایل‌های درایو‌هایی که از طریق SATA و SCSI به سیستم متصل شده‌اند، به صورت /dev/sdx می‌باشد. به طوری که X نشان دهنده‌ی هر درایو منحصر به فرد متصل شده به سیستم از طریق SATA یا SCSI می‌باشد و دقیقاً مثل قبل، اولین دستگاه متصل شده توسط SATA یا SCSI، نام sda را می‌گیرد، دومین دستگاه نام sdb را می‌گیرد و نام گذاری به همین ترتیب ادامه می‌یابد.

پارتیشن‌بندی درایوها

اکثر سیستم‌عامل‌های مدرن، به ما امکان پارتیشن‌بندی درایوها را می‌دهند. یک پارتیشن، قسمت مشخصی درون یک درایو است که سیستم عامل آن را به عنوان یک محل ذخیره‌سازی جداگانه در نظر می‌گیرد. پارتیشن‌بندی درایوها، مزیت‌های متعددی دارد که شامل:

• امکان نصب چند سیستم عامل

بسیاری از سیستم‌عامل‌ها برای نصب شدن روی کامپیوتر، احتیاج به یک پارتیشن منحصر به فرد دارند. ما با پارتیشن‌بندی، می‌توانیم روی یک درایو، چندین سیستم‌عامل داشته باشیم. البته با گسترش هر چه بیشتر مفهوم مجازی‌سازی، مفهوم وجود چند سیستم‌عامل روی یک سیستم، یا به عبارت دیگر، Dual Boot تا حد زیادی اهمیت خود را از دست داده است.

• تنوع فایل سیستم

با پارتیشن‌بندی یک درایو، می‌توانیم به هر پارتیشن، یک فایل سیستم متفاوت بدهیم. بعداً در مورد مفهوم فایل سیستم صحبت می‌کنیم، اما به طور کلی، فایل سیستم، به ساختمان داده‌هایی گفته می‌شود که چگونگی نگهداری و ذخیره‌ی اطلاعات روی یک پارتیشن را مدیریت می‌کنند. اهمیت وجود فایل سیستم‌های متفاوت روی یک درایو، زمانی مشخص می‌شود که تفاوت بین هر نوع فایل سیستم را بدانیم. مثلاً ممکن است یک فایل سیستم سریعتر از یک فایل سیستم دیگر باشد، یا ممکن است یک فایل سیستم محدودیت حجم فایل داشته باشد. مثلاً اگر با ویندوز کار کرده باشید، می‌دانید که فایل سیستم FAT، محدودیت حجم فایل دارد؛ یعنی ما نمی‌توانیم فایلی که حجمی فراتر از ۴ گیگابایت دارد را روی یک پارتیشن دارای فایل سیستم FAT بریزیم. این امر می‌تواند در بسیاری از مواقع برای ما مشکل‌ساز شود، پس باید به سراغ یک فایل سیستم دیگر برویم. از طرفی دیگر، برخی از دستگاه‌ها ممکن است نتوانند یک فایل سیستم خاص را بخوانند. مثلاً برخی از تلویزیون‌های قدیمی، نمی‌توانند فایل سیستمی به جز FAT را بخوانند. همه‌ی اینها، اهمیت پارتیشن‌بندی را نشان می‌دهد، چرا که با پارتیشن بندی می‌توانیم فایل سیستم هر پارتیشن را با توجه به نیازمان تغییر دهیم.

• مدیریت فضای درایو

با پارتیشن‌بندی درایو، می‌توانیم به هر پارتیشن یک حجم خاص بدهیم و بدین ترتیب، از پر شدن کل درایو و crash کردن سیستم جلوگیری کنیم. مدیریت فضای درایو یکی از کارهای مهم یک مدیر شبکه می‌باشد؛ چرا که اگر در پارتیشن‌بندی خود به درایوها به اندازه‌ی کافی حجم ندهیم، ممکن است سرویس‌های ما از کار بیافتند. همانطور که در جلسات قبل اشاره کردیم، همه‌ی سرویس‌های لینوکس

نظیر squid و... log های خود را در یک دایرکتوری مخصوص قرار می‌دهند. اگر یک سرویس به دلیل پر شدن فضای پارتیشن، نتواند چیزی در فایل‌های log خود بنویسد، از کار می‌افتد. این امر، اهمیت پارتیشن‌بندی در مدیریت فضا را نشان می‌دهد. بعداً می‌بینیم چطور با روش‌هایی نظیر LVM، می‌توانیم به صورت دینامیک حجم یک پارتیشن را کم یا زیاد کنیم.

• محافظت در برابر مشکلات درایو

درایوها گاهی اوقات به مشکلاتی بر می‌خورند. این مشکلات ممکن است ریشه‌ای سخت‌افزاری داشته باشند یا خطاهایی مربوط به فایل‌سیستم باشند. با پارتیشن‌بندی درایو، می‌توانیم مقیاس این مشکلات را کوچک‌تر کنیم. مثلاً اگر فایل‌سیستم یک پارتیشن دچار خرابی شود، به دلیل وجود چندین پارتیشن، اطلاعات موجود در پارتیشن‌های دیگر دچار مشکل نخواهند شد.

• امنیت

پارتیشن‌بندی به ما این امکان را می‌دهد که بتوانیم برای هر پارتیشن، permission تعریف کنیم. مثلاً ممکن است بخواهیم یک پارتیشن فقط به صورت read-only در سیستم وجود داشته باشد، یا بخواهیم یک پارتیشن فقط توسط یک سری از کاربران قابل خواندن باشد. با پارتیشن‌بندی، انجام این کار ساده می‌شود، چون در این حالت به جای تغییر permission‌های دسترسی به هر دایرکتوری و فایل، فقط کافی است دایرکتوری‌ها و فایل‌های مورد نظر را درون یک پارتیشن خاص قرار دهیم و permission را روی آن پارتیشن قرار دهیم.

• بک‌آپ

با پارتیشن‌بندی درایو، می‌توانیم از اطلاعات راحت‌تر بک‌آپ بگیریم. مثلاً می‌توانیم داده‌های مهم را در یک پارتیشن خاص قرار دهیم و فقط از آن پارتیشن بک‌آپ بگیریم.

سیستم‌های پارتیشن‌بندی

کامپیوترهایی که از BIOS برای اتصال به سخت‌افزار و بوت کردن سیستم‌عامل استفاده می‌کنند، مدیریت پارتیشن‌ها را با استفاده از سیستم Master Boot Record یا MBR انجام می‌دهند. این روش، روشی قدیمی می‌باشد و محدودیت‌های زیادی دارد که جلوتر با آنها آشنا می‌شویم. کامپیوترهای جدیدتر که از UEFI برای اتصال به سخت‌افزار و بوت کردن سیستم‌عامل استفاده می‌کنند، از سیستم GUID Partition Table یا GPT برای مدیریت پارتیشن‌ها استفاده می‌کنند. به طور کلی، هم MBR و هم GPT، اطلاعاتی نظیر این که هر پارتیشن از کدام سکتور شروع می‌شود و در کدام سکتور پایان می‌یابد، کدام پارتیشن bootable (قابل بوت شدن) می‌باشد و... را در اختیار سیستم قرار می‌دهند.

در این بخش، سعی می‌کنیم MBR و GPT را به صورت مختصر توضیح دهیم و در مورد مزیت‌ها و محدودیت‌های آنها صحبت کنیم.

MBR

روش MBR در سال ۱۹۸۳ همراه با سیستم‌عامل PC DOS 2.0 به عنوان سیستمی برای مدیریت پارتیشن‌ها معرفی شد. MBR کلیه داده‌های خود را بر روی اولین سکتور هارد دیسک قرار می‌دهد. این داده‌ها، شامل

بوت‌لودر سیستم‌عامل و همچنین اطلاعات مربوط به پارتیشن‌بندی سیستم می‌باشند.

نکته: جلسه‌ی قبل در مورد بوت‌لودرها صحبت کردیم، اما به طور کلی، بوت‌لودری که روی اولین سکتور هارددیسک قرار می‌گیرد، یک برنامه‌ی کوچک است که هنگام اجرا، یک بوت‌لودر بزرگ‌تر واقع در یک پارتیشن دیگر را فرا می‌خواند.

اگر دقت کنید، گفتیم که MBR، داده‌های خود را روی اولین سکتور هارد دیسک قرار می‌دهد. هر سکتور هارددیسک، ۵۱۲ بایت حجم دارد. در واقع MBR باید بوت‌لودر و همچنین اطلاعات مربوط به پارتیشن‌بندی سیستم را روی ۵۱۲ بایت فضا جا دهد. در MBR، ۶۴ بایت از فضای اولین سکتور، به اطلاعات پارتیشن‌بندی اختصاص دارد و هر پارتیشن، به ۱۶ بایت فضا برای ذخیره‌ی اطلاعات خود نیاز دارد. این یعنی ما در سیستم MBR، فقط می‌توانیم ۴ پارتیشن داشته باشیم. به این پارتیشن‌ها، پارتیشن‌های اصلی یا Primary Partitions می‌گویند.

یکی از مشکلات MBR، همین امر می‌باشد؛ در محیط کاری، خیلی از کاربران به بیشتر از ۴ پارتیشن نیاز دارند و این محدودیت MBR برای آنها مشکل‌ساز می‌باشد.

برای رفع این مشکل، از مفهومی به نام Extended Partition یا «پارتیشن قابل تعمیم» استفاده می‌کنیم. MBR می‌تواند یکی از این چهار پارتیشن اصلی را تبدیل به Extended Partition کند. Extended Partition، به ما اجازه می‌دهد که تعدادی بی‌نهایت پارتیشن منطقی یا Logical Partitions داشته باشیم. برای درک بهتر این امر، بیایید به ساختار کلی ساختار MBR نگاهی بیاندازیم:

پارتیشن اصلی چهارم / پارتیشن Extended			پارتیشن اصلی سوم	پارتیشن اصلی دوم	پارتیشن اصلی اول
پارتیشن منطقی	پارتیشن منطقی	پارتیشن منطقی			

تصویر ۱- ساختار کلی MBR

همانطور که بالاتر گفتیم و در تصویر ۱ می‌بینید، ما می‌توانیم چهار پارتیشن اصلی داشته باشیم که یکی از آنها می‌تواند تبدیل به Extended Partition شود و آن Extended Partition، می‌تواند تعدادی بی‌نهایت پارتیشن منطقی درون خود داشته باشد. در تصویر ۱، ما شش پارتیشن داریم که سه تا از آنها منطقی می‌باشند. ما نمی‌توانیم تعداد پارتیشن‌های اصلی را افزایش دهیم، اما می‌توانیم تعداد پارتیشن‌های منطقی را تا نفس و حوصله داریم، افزایش دهیم.

پس به طور کلی، می‌توانیم ویژگی‌های MBR را به صورت زیر بیان کنیم:

- همانطور که گفتیم روی یک دیسک MBR، فقط می‌توان ۴ پارتیشن اصلی داشت. هارددیسک‌ها از چندین هزار سکتور تشکیل شده‌اند و هر سکتور، حجمی حدود ۵۱۲ بایت دارد. در MBR، ۶۴ بایت موجود در سکتور اول هارددیسک، به ذخیره‌ی اطلاعات پارتیشن‌ها اختصاص دارد و اطلاعات هر پارتیشن، ۱۶ بایت فضا می‌گیرد، به همین دلیل، ما فقط می‌توانیم ۴ پارتیشن اصلی داشته باشیم.
- در MBR، ما ۳ نوع پارتیشن داریم: پارتیشن اصلی، پارتیشن Extended و پارتیشن منطقی. پارتیشن‌های Extended و منطقی برای دور زدن محدودیت ۴ پارتیشنی MBR به وجود آمده‌اند.

۳. اطلاعاتی که در ۱۶ بایت مخصوص هر پارتیشن ذخیره می‌شود، شامل شناسه‌ی فایل سیستم، نقطه‌ی شروع و پایان پارتیشن، شماره‌ی سیلندر ابتدایی و پایانی و... می‌باشد.

۴. در دیسک‌های MBR، نمی‌توانیم پارتیشنی با حجم فراتر از ۲ ترابایت (یا به صورت دقیق‌تر، ۲ تیبایت، معادل ۲,۲ ترابایت) داشته باشیم.

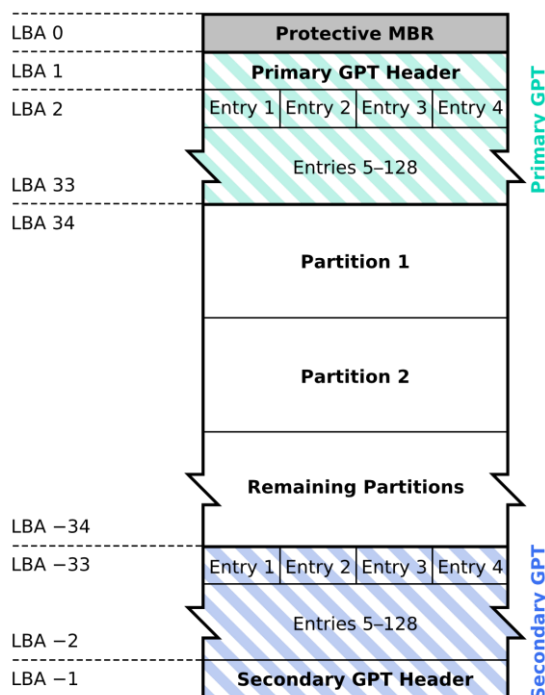
در MBR، شماره‌گذاری پارتیشن‌های اصلی از ۱ تا ۴ می‌باشد و شماره‌گذاری پارتیشن‌های منطقی از ۵ شروع می‌شود. در شماره‌گذاری پارتیشن‌های اصلی، می‌توان فاصله‌ی خالی داشت، مثلاً می‌توان شماره‌ی پارتیشن اول را ۱ و پارتیشن دوم را ۳ گذاشت، اما شماره‌گذاری پارتیشن‌های منطقی باید کاملاً پشت سر هم صورت گیرد. به عبارت دیگر، ما می‌توانیم یک هارددرایو با شماره پارتیشن ۱، ۳، ۵ و ۷ داشته باشیم، اما نمی‌توانیم یک هارددرایو با شماره پارتیشن ۱، ۳، ۵ و ۷ داشته باشیم. اگر پارتیشن ۷ وجود داشته باشد، حتماً باید پارتیشن ۵ و ۶ نیز وجود داشته باشند.

همانطور که گفتیم، MBR علاوه بر اطلاعات پارتیشن، بوت‌لودر سیستم‌عامل را نیز درون خود نگهداری می‌کند. این امر حساسیت و اهمیت بالای MBR را نشان می‌دهد؛ از آنجایی که MBR روی اولین سکتور هارددرایو وجود دارد، احتمال وجود آمدن خطا در آن نیز زیاد است، پس وجود بک‌آپ از اهمیت بیشتری برخوردار می‌شود.

GPT

GPT سیستم جدیدتری برای پارتیشن‌بندی درایوها می‌باشد و به آهستگی در حال جایگزینی MBR است. GPT بسیار پیچیده‌تر از MBR می‌باشد. برای درک بهتر عملکرد GPT، به تصویر ۲ توجه کنید. GPT به جای تقسیم‌بندی دیسک به سکتورها، آنها را به بلوک‌ها تقسیم می‌کند.

GUID Partition Table Scheme



تصویر ۲- نمایی از طرح‌بندی سیستم GPT

تصویر ۲، ساختار یک دیسک GPT را نشان می‌دهد. همانطور که در تصویر می‌بینید، آدرس‌دهی بلوک‌ها در GPT با پیشوند LBA شروع می‌شود. LBA که مخفف Logical Block Addressing می‌باشد، یک روش برای آدرس‌دهی پارتیشن‌ها در دستگاه‌های ذخیره‌سازی بزرگ نظیر هارددیسک‌ها و SSDها می‌باشد. به عبارت دیگر، LBA 0 نشان دهنده‌ی اولین بلوک موجود در در دیسک، LBA 1 نشان دهنده‌ی دومین بلوک و به همین ترتیب، می‌باشد. اگر درک مفهوم بلوک برایتان دشوار است، می‌توانید بلوک را معادل سکتور در نظر بگیرید. البته از در واقعیت بلوک و سکتور با هم متفاوت می‌باشند، اما چون درس ما سخت‌افزار نیست، زیاد درگیر جزئیات نمی‌شویم.

همانطور که در تصویر ۲ می‌بینید، LBA 0، برای MBR کنار گذاشته شده است. دلیل وجود این بلوک، compatibility یا سازگاری با نرم‌افزارها و سیستم‌های قدیمی می‌باشد و به طور کلی، عملکرد MBR که در بخش قبل گفتیم را ندارد. در LBA 1 که Primary GPT Header نام دارد، پوینترهایی به جدول پارتیشن GPT وجود دارد. در این قسمت، بلوک‌های قابل استفاده در دیسک و همچنین تعداد و سایز پارتیشن‌های موجود در جدول پارتیشن مشخص شده‌اند.

LBA 3، جدول پارتیشن را درون خود دارد. در GPT، اطلاعات هر پارتیشن، ۱۲۸ بایت فضا می‌گیرد. پس در هر بلوک، می‌توانیم ۴ پارتیشن داشته باشیم. همانطور که می‌بینید، در مقایسه با MBR، سیستم GPT به فضای بیشتری برای هر پارتیشن نیاز دارد. این فضای اضافه‌تر، باعث می‌شود که بتوانیم در جدول پارتیشن، اطلاعات بیشتری در مورد هر پارتیشن ذخیره کنیم. در سیستم‌های ویندوزی جدول پارتیشن GPT تعداد ۳۲ بلوک را از ما می‌گیرد، این یعنی که به صورت پیش‌فرض می‌توانیم ۱۲۸ پارتیشن داشته باشیم. البته در بسیاری از اوقات به این تعداد پارتیشن نیاز نداریم، اما این تعداد به صورت رزرو در GPT وجود دارد. نکته‌ی قابل توجه در تصویر، وجود LBA -1 تا LBA -33 در سیستم GPT، می‌باشد. این بلوک‌ها، نشان دهنده‌ی آخرین بلوک‌های موجود روی هارددیسک می‌باشند و به عنوان بک‌آپ برای GPT Header و جدول پارتیشن عمل می‌کنند.

پس به طور خلاصه، می‌توان گفت که GPT محدودیت‌های MBR را ندارد و به نوعی می‌تواند بی‌نهایت پارتیشن داشته باشد. علاوه بر این، GPT محدودیت حجمی نیز ندارد، یعنی بر خلاف MBR، می‌توان پارتیشن‌هایی تا سایز ۹,۴ زتابایت داشت. بر خلاف MBR که اطلاعات مربوط به پارتیشن‌ها و همچنین بوت‌لودر را در یک مکان قرار می‌داد و به نوعی سیستم را مستعد خرابی می‌کرد؛ GPT چندین کپی از اطلاعات پارتیشن و بوت را درون دیسک نگهداری می‌کند و علاوه بر این، GPT از CRC جهت سنجش یکپارچگی اطلاعات موجود بر روی دیسک استفاده می‌کند و در صورت پیدا کردن خطا، سعی به ترمیم اطلاعات می‌کند. شاید بتوان جدی‌ترین مشکل GPT را عدم سازگاری با سیستم‌هایی که از طریق BIOS بوت می‌شوند دانست، چرا که GPT فقط با سیستم‌های UEFI سازگاری دارد.

تشخیص اتوماتیک درایوها

سیستم‌های لینوکس درایوها و پارتیشن‌ها را هنگام بوت شدن تشخیص داده و به هر کدام یک دیوایس‌فایل خاص در /dev اختصاص می‌دهند. با محبوبیت هر چه بیشتر درایوهای USB (مثل فلش مموری‌ها، هاردهای اکسترنال و...) که در هر لحظه می‌توانند به سیستم متصل شوند، سیستم‌های لینوکس باید چگونگی تشخیص درایوها را تغییر می‌دادند.



امروزه اکثر سیستم‌های لینوکسی از برنامه‌ی udev برای تشخیص درایوها استفاده می‌کنند. udev برنامه‌ای است که به صورت مداوم در پشت‌صحنه‌ی سیستم در حال اجرا می‌باشد و به صورت اتوماتیک، هر سخت‌افزاری که به یک سیستم لینوکسی در حالا چرا متصل شود را تشخیص می‌دهد. به محض وصل کردن یک درایو جدید، udev آنها را تشخیص داده و به هر کدام یک دیوایس فایل جدید و منحصر به فرد در دایرکتوری /dev اختصاص می‌دهد.

یکی از ویژگی‌های جالب udev، ایجاد دیوایس فایل‌های مستدام (persistent) برای هر درایو می‌باشد. وقتی که ما یک درایو را به سیستم وصل می‌کنیم، نام دیوایس فایل موجود در /dev، می‌تواند با توجه به تعداد درایوهای وصل شده به سیستم، ترتیب اتصال آنها و... دچار تغییر شود. این امر می‌تواند برنامه‌های متفاوتی که از یک درایو خاص استفاده می‌کنند را دچار مشکل کند. برای حل این مشکل، برنامه‌ی udev با استفاده از ویژگی‌های منحصر به فرد هر درایو، لینک‌هایی به دیوایس فایل‌های موجود در /dev را در دایرکتوری /dev/disks قرار می‌دهد. به این لینک‌ها، device link می‌گویند. udev چهار دایرکتوری متفاوت برای ایجاد این لینک‌ها درست می‌کند:

- /dev/disk/by-id: درایوها را با توجه به نام شرکت سازنده، مدل و شماره سریال به دیوایس فایل مربوطه لینک می‌کند.
- /dev/disk/by-label: درایوها را با توجه به برچسب‌شان به دیوایس فایل مربوطه لینک می‌کند.
- /dev/disk/by-path: درایوها را با توجه به شماره پورتی که به آن متصل شده‌اند به دیوایس فایل مربوطه لینک می‌کند.
- /dev/disk/by-uuid: درایوها را با توجه به شماره‌ی UUID هر دستگاه، به دیوایس فایل مربوطه لینک می‌کند.

با استفاده از دیوایس لینک‌های udev، می‌توانیم از یک درایو خاص، با استفاده از شناسه‌ی منحصر به فرد آن استفاده کنیم تا مجبور به استفاده از دیوایس فایل‌هایی که نامشان بسته به زمان متصل کردن درایو به سیستم تغییر می‌کند، نباشیم.

Mount Point ها

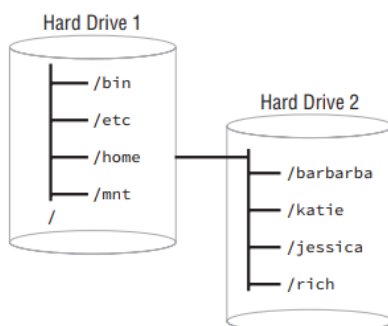
گفتیم که وقتی یک هارددیسک به سیستم وصل می‌شود، سیستم‌عامل دیوایس فایل مربوط به آن را در دایرکتوری /dev قرار می‌دهد. حال سیستم‌عامل باید بتواند به طریقی به محتویات موجود در آن و همچنین پارتیشن‌های آن دسترسی پیدا کند. در سیستم‌عامل ویندوز، driver letter نامی نظیر C:، D: و... به سیستم‌عامل امکان دسترسی به محتویات پارتیشن‌ها را می‌دهند. در لینوکس، هر پارتیشن در یک دایرکتوری قرار می‌گیرد. یعنی برای دسترسی به آن پارتیشن، به جای رفتن به یک نقطه‌ای به نام درایو C:، ما باید وارد یک دایرکتوری شویم. به دایرکتوری‌هایی که یک پارتیشن در آنها قرار گرفته است، mount point می‌گویند و به عمل قرار دادن یک پارتیشن درون یک دایرکتوری، mount کردن می‌گویند. پس mount point، صرفاً یک دایرکتوری است که از آن برای دسترسی به پارتیشن مورد نظر استفاده می‌کنیم و با mount کردن یک پارتیشن، آن پارتیشن را به آن mount point، لینک می‌کنیم.

به عبارت دیگر، ما دیوایس فایل هر هارددیسک یا پارتیشن را، بر روی یک mount point، mount می‌کنیم.

همانطور که قبلا نیز به آن اشاره کردیم، ساختار دایرکتوری‌ها در لینوکس با یک دایرکتوری پایه، به نام دایرکتوری root (/) آغاز می‌شود. دایرکتوری روت، فایل‌ها و سایر دایرکتوری‌های موجود درون خود را با توجه به مسیری که برای رسیدن به آنها باید طی شود، لیست می‌کند. برای مثال، مسیر یک فایل در لینوکس، نامی نظیر زیر خواهد داشت:

/etc/postfix/main.cf

همانطور که می‌بینید، مسیر یک فایل در لینوکس بسیار متفاوت با مسیر یک فایل در ویندوز می‌باشد. مثلا در ویندوز برای نوشتن مسیر یک فایل، می‌نویسیم C:\Program Files\test.txt، اما همانطور که گفتیم، در لینوکس چیزی به اسم drive letter نداریم. اگر به مسیر فایل بالا نگاه کنید، می‌بینید که مسیر به ما می‌گوید فایل main.cf در دایرکتوری postfix در دایرکتوری /etc قرار دارد؛ به عبارت دیگر، از این مسیر، نمی‌توانیم بفهمیم که این فایل روی کدام هارددیسک یا حتی پارتیشن، قرار دارد.



تصویر ۳- بررسی مفهوم مانت پوینت‌ها (Mount Point)

همانطور که در تصویر ۳ می‌بینید، ما در اینجا دو هارددیسک متفاوت داریم. یکی از این هارددیسک‌ها دایرکتوری اصلی، یعنی دایرکتوری روت (/) را درون خود دارد و هارددیسک دوم، دایرکتوری‌های شخصی هر کاربر را درون خود دارد. زمانی که هارددیسک دوم روی سیستم قرار گیرد، کلیه فایل‌ها و دایرکتوری‌های موجود در آن هارددیسک، از دایرکتوری /home قابل دسترس خواهند بود. در واقع در تصویر ۳، هارددیسک دوم، بر روی دایرکتوری /home موجود بر روی هارددیسک اول، mount شده است؛ یا به صورت فنی‌تر، mount point هارددیسک دوم، دایرکتوری /home می‌باشد.

از آنجا که لینوکس همه چیز را تحت دایرکتوری روت قرار می‌دهد، مدیریت این دایرکتوری‌ها و فایل‌هایی که در آن قرار می‌گیرند کمی پیچیده می‌شود. اگر به خاطر داشته باشید، دو جلسه پیش در مورد FHS یا Filesystem Hierarchy Standard صحبت کردیم. FHS استاندارد است که دایرکتوری‌های اصلی که باید روی هر سیستم لینوکسی وجود داشته باشند و همچنین نوع اطلاعاتی که باید در هر دایرکتوری قرار گیرد را مشخص می‌کند.

اغلب، با توجه به هدف ما از نصب یک سیستم لینوکسی (استفاده از آن به عنوان وب سرور، میل سرور و...) ممکن است هر کدام از دایرکتوری‌ها به عنوان یک پارتیشن متفاوت در نظر گرفته شوند. در جدول ۱، برخی از دایرکتوری‌های مشخص شده توسط FHS، وظیفه‌ی هر کدام و همچنین میزان حجمی که به هر دایرکتوری باید اختصاص داده شود را می‌بینید.

نکته: برخی از دایرکتوری‌ها نظیر /etc, /bin, /sbin, /lib و /dev هرگز نباید در یک پارتیشن جداگانه قرار گیرند؛ چرا که این دایرکتوری‌ها اطلاعات مهم سیستمی را درون خود دارند و بدون آنها، لینوکس نمی‌تواند به

جدول ۱ - برخی از دایرکتوری‌های مشخص شده توسط FHS و عملکرد و حجم پیشنهادی آنها

دایرکتوری	عملکرد	حجم اختصاصی
/swap	یک پارتیشن جداگانه روی هارد دیسک که در کنار RAM کار می‌کند. سرعت آن بسیار کمتر از RAM می‌باشد اما به سیستم اجازه می‌دهد که برنامه‌های بیشتر و بزرگتری را اجرا کند.	دو یا سه برابر حجم RAM
/boot	فایل‌های مخصوص بوت سیستم را درون خود نگهداری می‌کند.	۱ گیگابایت
/home	فایل‌های مخصوص هر کاربر در سیستم را درون خود نگهداری می‌کند. اختصاص یک پارتیشن جداگانه به این دایرکتوری، باعث می‌شود اطلاعات کاربران پس از آپگرید سیستم دست‌نخورده بماند.	بستگی به تعداد کاربران دارد. می‌تواند از ۲۰۰ مگابایت تا ۳ ترابایت و بیشتر باشد.
/opt	برنامه‌های لینوکسی که Proprietary هستند (این سورس نیستند) در این دایرکتوری قرار می‌گیرند.	۱۰۰ مگابایت الی ۵ گیگابایت
/tmp	فایل‌های temp را درون خود نگهداری می‌کند.	۱۰۰ مگابایت الی ۲۰ گیگابایت
/usr	برنامه‌های لینوکس و برخی از داده‌های آنها در این دایرکتوری قرار می‌گیرند.	۵۰۰ مگابایت الی ۲۵ گیگابایت
/usr/local	برنامه‌های لینوکسی که خودمان کامپایل می‌کنیم در این دایرکتوری قرار می‌گیرد.	۱۰۰ مگابایت الی ۳ گیگابایت
/var	فایل‌های متفاوتی که سیستم در زندگی روزمره (!) خود تولید می‌کند در این دایرکتوری قرار می‌گیرد. فایل‌هایی نظیر logها و... معمولا در این دایرکتوری قرار می‌گیرند. این دایرکتوری در اکثر اوقات، مخصوصا وقتی که سیستم به عنوان میل سرور، cache server و... به کار برده شود، در یک پارتیشن جداگانه قرار می‌گیرد.	۱۰۰ مگابایت الی ۳ ترابایت

پارتیشن‌بندی هنگام نصب سیستم

تا به اینجا زیاد در مورد پارتیشن‌بندی صحبت کرده‌ایم، اما چگونه و در چه زمانی باید هارد دیسک‌ها را پارتیشن‌بندی کنیم؟ خیلی از اوقات، یک سیستم را با توجه به کاربرد آن، هنگام نصب پارتیشن‌بندی می‌کنند؛ چرا که پارتیشن‌بندی یک فایل‌سرور با پارتیشن‌بندی یک DNS سرور باید کاملا متفاوت باشد. ما در این بخش، می‌خواهیم در مورد چگونگی پارتیشن‌بندی هنگام نصب صحبت کنیم.

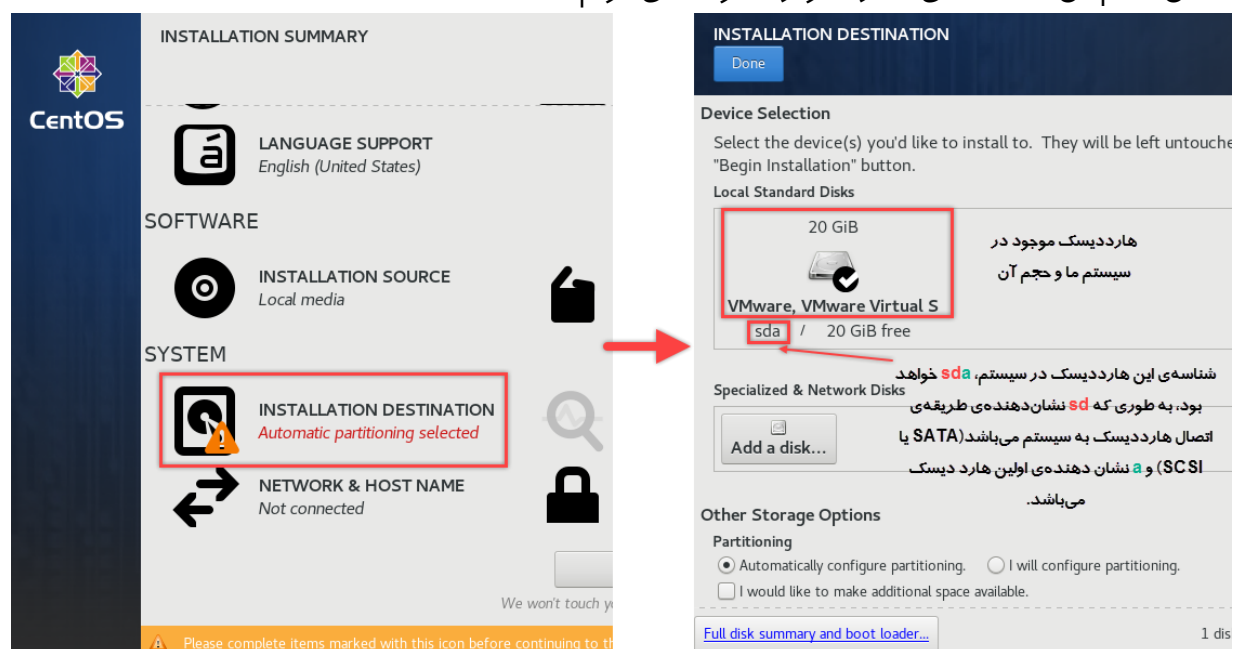


در جزوه‌ی جلسه‌ی اول، به طور مفصل در مورد چگونگی نصب لینوکس در یک ماشین مجازی صحبت کردیم. الان می‌خواهیم پارتیشن‌بندی را هنگام نصب یک لینوکس جدید انجام دهیم. پس یک ماشین مجازی جدید ایجاد کنید و مراحل نصب را تا رسیدن به تصویر ۴، ادامه دهید.



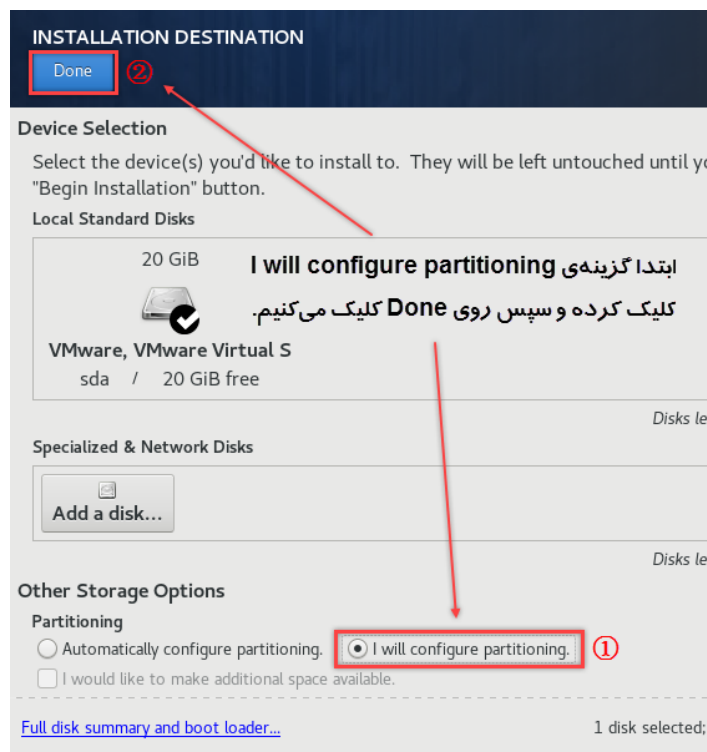
تصویر ۴- تنظیمات نصب یک سیستم CentOS جدید

برای پارتیشن‌بندی، ما باید از قسمت SYSTEM بر روی INSTALLATION DESTINATION کلیک کنیم. به محض انجام این کار، با نمایی نظیر تصویر ۵ مواجه می‌شویم:



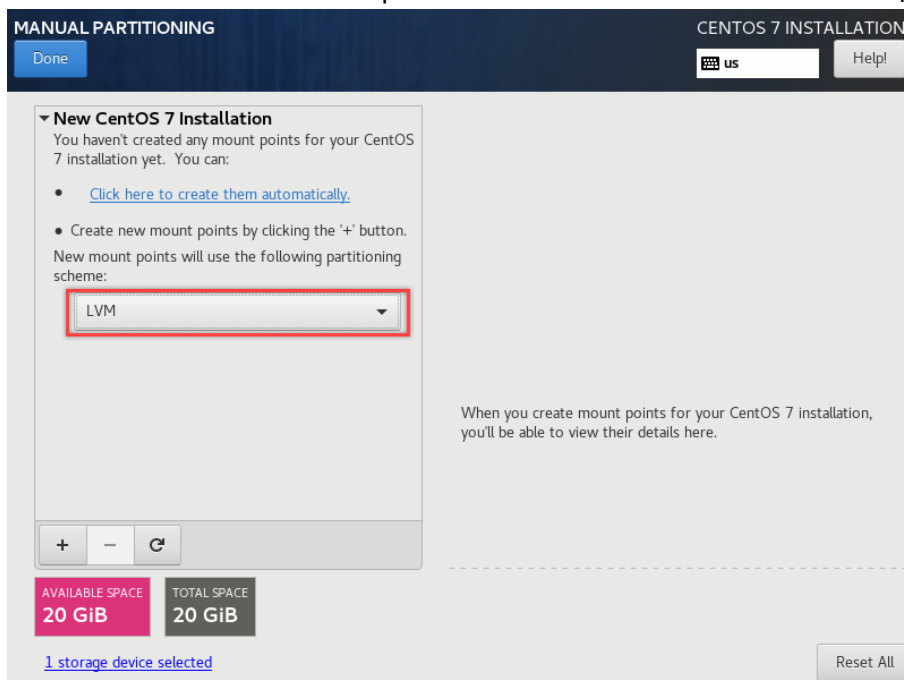
تصویر ۵- رفتن به بخش تنظیمات هارددیسک و پارتیشن‌بندی

در صفحه‌ی ظاهر شده، روی گزینه‌ی I will configure partitioning کلیک کرده و سپس دکمه‌ی Done را می‌زنیم:



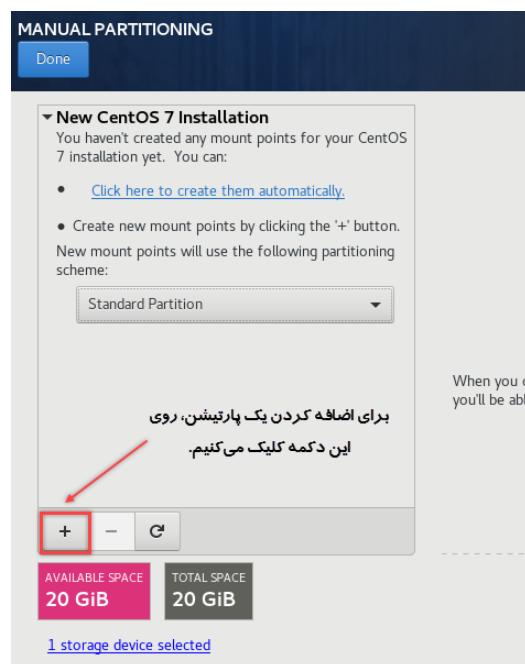
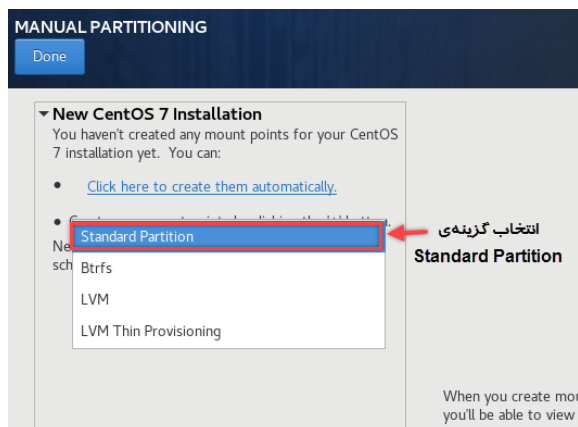
تصویر ۶- انتخاب گزینه‌ی مشخص شده جهت ورود به صفحه‌ی پارتیشن‌بندی

به محض انجام این کار، با نمایشی نظیر تصویر ؟ مواجه می‌شویم:



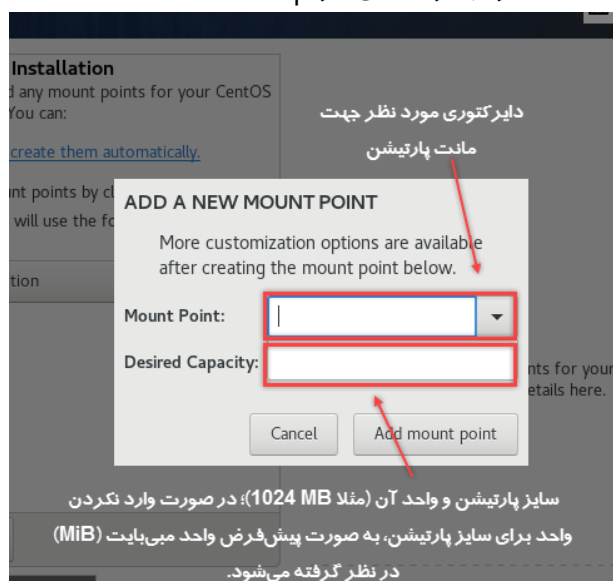
تصویر ۷- نمایشی از صفحه‌ی پارتیشن‌بندی

همانطور که می‌بینید، CentOS به صورت پیش فرض می‌خواهد پارتیشن‌بندی ما را به صورت LVM انجام دهد. بعداً در مورد LVM صحبت خواهیم کرد، اما فعلاً از این بخش، گزینه‌ی Standard Partitioning را انتخاب می‌کنیم و سپس روی دکمه‌ی + کلیک می‌کنیم.



تصویر ۸- انتخاب گزینه‌ی Standard Partition جهت پارتیشن‌بندی به صورت استاندارد

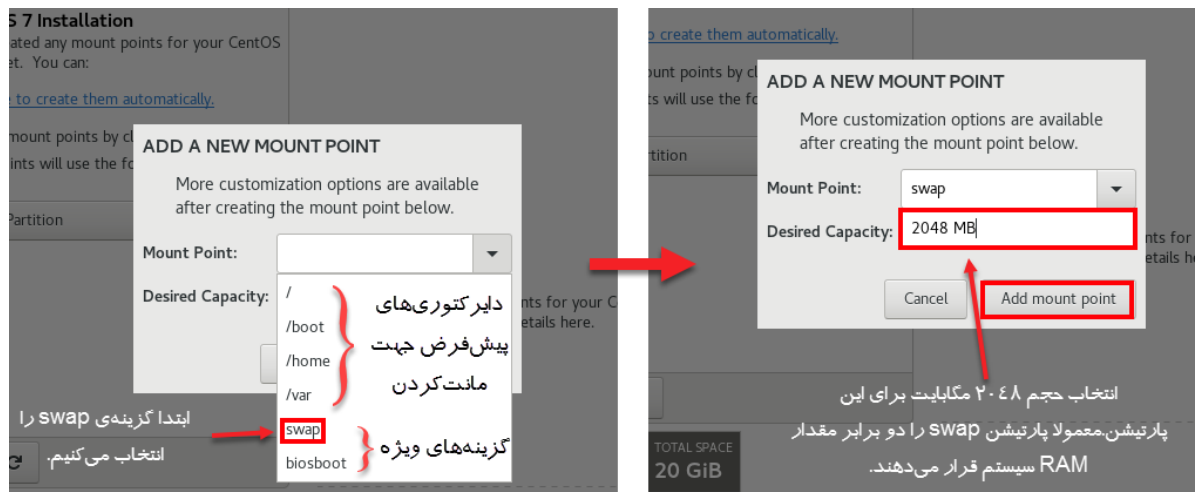
به محض کلیک روی دکمه‌ی +، با تصویر مواجه می‌شویم:



تصویر ۹- پنجره‌ی مربوط به ایجاد یک پارتیشن جدید

همانطور که می‌بینید، در اینجا می‌توانیم Mount Point مورد نظر جهت قرار گیری پارتیشن و همچنین سایز پارتیشن به می‌بایست را وارد کنیم. در قسمت Mount Point، می‌توانیم هر کدام از دایرکتوری‌های عنوان شده در جدول ۱ را قرار دهیم و حتی می‌توانیم دایرکتوری دلخواه خود را ایجاد کنیم. به عبارت دیگر ما در این پنجره، به سیستم می‌گوییم که پارتیشن ما چه حجمی داشته باشد و همچنین در چه Mount Point یا دایرکتوری مانیت شود.

در این پنجره، یک سری Mount Point یا دایرکتوری پیش‌فرض جهت مانیت پارتیشن قرار دارد که بهتر است ابتدا آنها را بررسی کنیم. اگر روی فلش رو به پایین در قسمت Mount Point کلیک کنید، چندین دایرکتوری و دو عنوان خاص می‌بینید. یکی از این عناوین خاص، swap می‌باشد. بیاید ابتدا یک پارتیشن برای swap ایجاد کنیم. برای این کار در قسمت Mount Point، گزینه‌ی swap را انتخاب می‌کنیم و سپس مقداری به آن می‌دهیم:



تصویر ۱۰ - ایجاد یک پارتیشن برای swap

پس از مشخص کردن ظرفیت پارتیشن swap، روی دکمه Add mount point کلیک می‌کنیم و با صفحه‌ای نظیر تصویر ۱۱ مواجه می‌شویم:

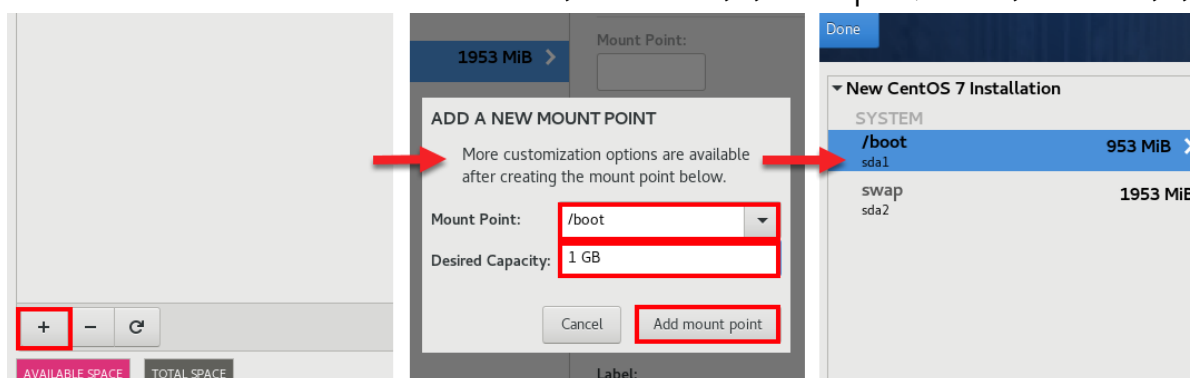


تصویر ۱۱ - نمایش از همه‌ی پارتیشن‌های ایجاد شده تا به اینجا

همانطور که می‌بینید، پارتیشن swap ما ایجاد شده و ۱۹۵۳ مبی‌بایت، معادل ۲۰۴۸ مگابایت ظرفیت دارد. توجه کنید که در این بخش، CentOS به صورت اتوماتیک ظرفیت مشخص شده توسط شما را در واحد مبی‌بایت نشان می‌دهد.

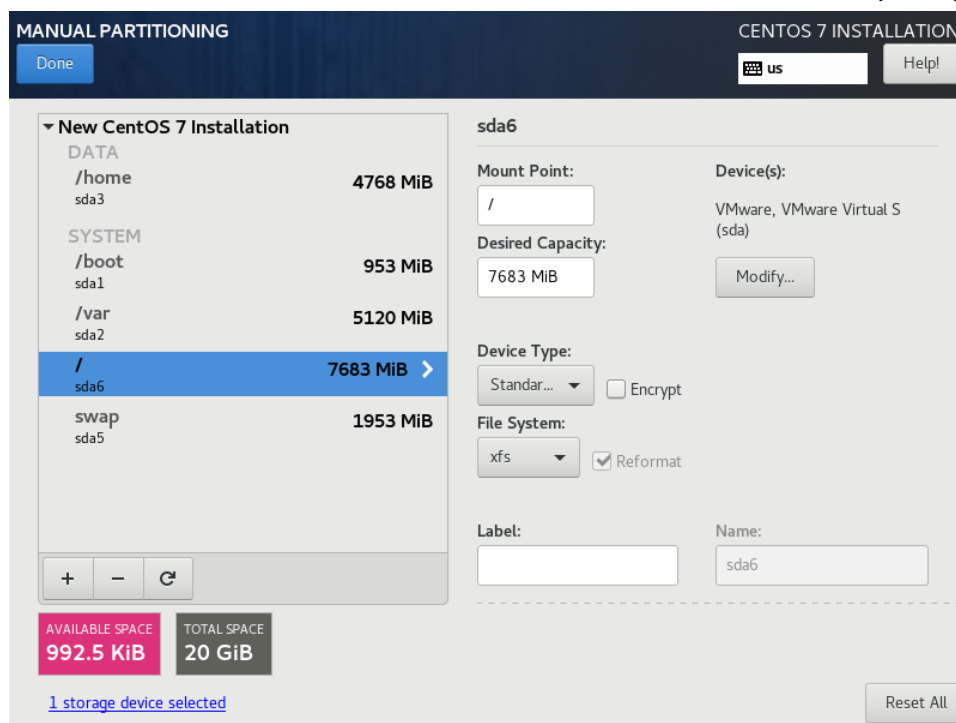
اگر کمی بیشتر توجه کنید، می‌بینید که پارتیشن swap نام sda1 را دارد. sda1، یعنی اولین پارتیشن بر روی اولین هارددرایوی که از طریق SCSI یا SATA به سیستم متصل شده است. پارتیشن بعدی روی این دستگاه، sda2 نام می‌گیرد، پارتیشن بعد از آن sda3 و به همین ترتیب. اگر یک هارددرایو دیگر را از طریق SCSI یا SATA به سیستم متصل کنیم، نام آن sdb می‌شود و پارتیشن اول آن sdb1 نام می‌گیرد و به همین ترتیب.

در تصویر ۱۱، در بخش Device Type، گزینه‌ی Standard Partitioning انتخاب شده است. این به معنای پارتیشن‌بندی دستگاه‌ها به روش کلاسیک می‌باشد. ما می‌توانیم از LVM نیز برای پارتیشن‌بندی استفاده کنیم، اما بحث در مورد LVM را به بعد می‌سپاریم. با انتخاب گزینه‌ی Encrypt، می‌توانیم این پارتیشن را رمزگذاری کنیم؛ اما رمزگذاری کردن پارتیشن swap منطقی نیست. همانطور که گفتیم، swap مکانی است که سیستم اطلاعات خود را هنگام پر بودن RAM در آن قرار می‌دهد، پس رمزگذاری آن کاری بی‌فایده می‌باشد. در نهایت در بخش Filesystem نیز باید گزینه‌ی swap انتخاب شود. فعلاً نگران این بخش و مفهوم آن نباشید، چون بعداً به صورت مفصل در مورد فایل‌سیستم‌ها صحبت می‌کنیم. خوب، بیاید یک پارتیشن دیگر نیز به سیستم اضافه کنیم. پس روی دکمه‌ی + کلیک می‌کنید، Mount Point را برابر با /boot قرار داده و حجم آن را برابر ۱ گیگابایت قرار دهید:



تصویر ۱۲ - ایجاد پارتیشن بوت

ایجاد پارتیشن‌های /home، /var و /، به علاوه‌ی هر پارتیشن دیگر را به خودتان می‌سپاریم. توجه کنید که برای این که بتوانید سیستم را نصب کنید، حتماً باید پارتیشن / را ایجاد کنید. در پایان کار پارتیشن‌بندی، با تصویری نظیر زیر مواجه خواهید شد:



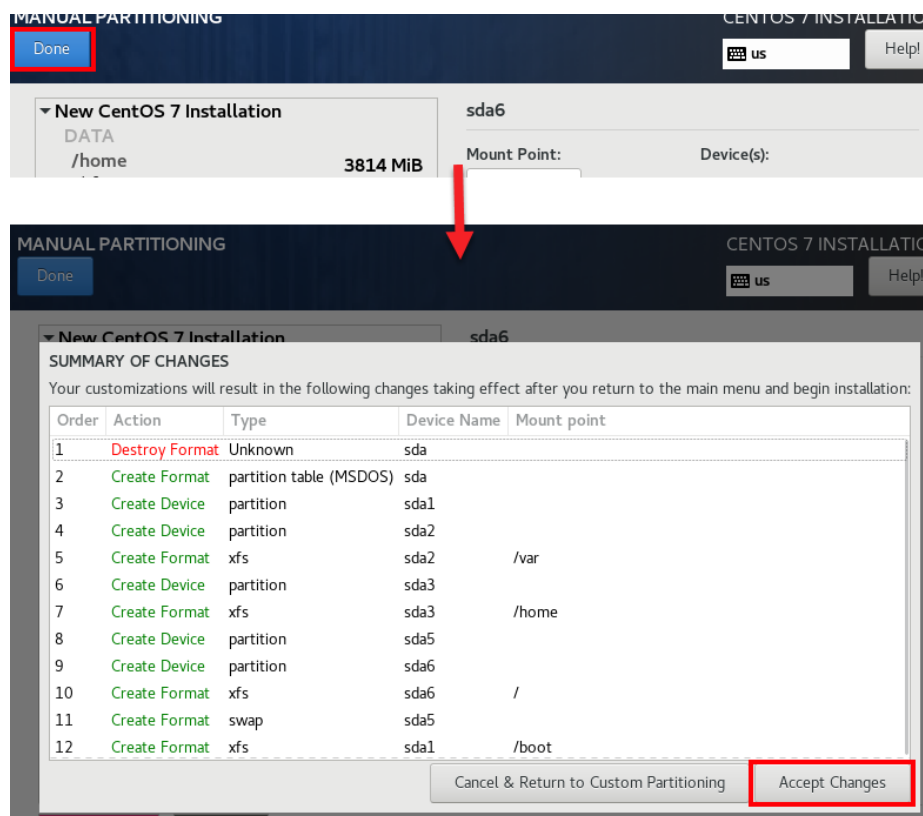
تصویر ۱۳ - نمایی از پارتیشن‌های ایجاد شده برای این سیستم

همانطور که در تصویر بالا می‌بینید، نام پارتیشن‌ها دچار تغییرهایی شده‌اند. مثلاً swap که در ابتدا sda1 بود، اکنون تبدیل به sda5 شده است و sda1 به پارتیشن /boot اختصاص یافته است. دلیل این امر، این است که اولین پارتیشن ما، باید پارتیشن /boot باشد. در پارتیشن /boot، بوت‌لودر اصلی سیستم قرار گرفته است (بوت‌لودری که توسط بوت‌لودر کوچک‌تر موجود در MBR اجرا می‌شود) و به همین دلیل، پارتیشن /boot با نام sda1 در سیستم قرار می‌گیرد.

نکته‌ی قابل توجه دیگر، عدم وجود sda4 می‌باشد. برای درک دلیل عدم وجود این پارتیشن، باید به بحثمان در مورد سیستم‌های متفاوت پارتیشن‌بندی بازگردیم. در حال حاضر، ما سیستم را با روش MBR پارتیشن‌بندی می‌کنیم. همانطور که گفتیم، در MBR می‌توانیم چهار پارتیشن اصلی داشته باشیم. اگر بخواهیم بیشتر از ۴ پارتیشن داشته باشیم، باید یکی از پارتیشن‌ها را تبدیل به Extended Partition کنیم و تعدادی پارتیشن منطقی درون آن بسازیم. این دقیقاً کاری است که در اینجا انجام شده است. ما در این سیستم، ۵ پارتیشن ایجاد کرده‌ایم. چون MBR به ما اجازه‌ی داشتن ۵ پارتیشن اصلی را نمی‌دهد، پارتیشن چهارم (sda4)، تبدیل به یک Extended Partition شده و ما درون آن، دو پارتیشن منطقی (sda5 و sda6) ایجاد کرده‌ایم. به همین دلیل، ما پارتیشنی با نام sda4 نمی‌بینیم، چون در واقع sda4 تقسیم به چند پارتیشن منطقی شده است.

نکته: در پارتیشن‌بندی، حتماً باید پارتیشن‌های / و /boot را ایجاد کنیم، وگرنه نمی‌توانیم سیستم‌عامل را نصب کنیم. همچنین داشتن پارتیشن swap شدیداً پیشنهاد می‌شود. اگر پارتیشن swap ایجاد نکنید، سیستم‌عامل به شما هشدار می‌دهد، اما جلوی نصب سیستم‌عامل را نمی‌گیرد.

حال برای نصب سیستم‌عامل، کافی است روی دکمه‌ی Done کلیک کرده و سپس دکمه‌ی Accept Changes را بزنید:



تصویر ۱۴ - تایید پارتیشن‌بندی ایجاد شده

پس از این کار، به صفحه‌ی موجود در تصویر ۴ بازگردانده می‌شوید. کافی است بر روی Begin Installation کلیک کنید تا سیستم جدید شما نصب شود. حین نصب، باید یک رمز برای کاربر root ایجاد کنید. ما این امر را به طور مفصل در جزوه جلسه اول توضیح دادیم، پس در اینجا به آن نمی‌پردازیم. پس از پایان نصب، ماشین مجازی را Reboot کنید.

مشاهده‌ی اطلاعات پارتیشن‌بندی

خیلی از اوقات خود ما پارتیشن‌بندی سیستم را انجام نداده‌ایم و نیاز داریم تا اطلاعات پارتیشن‌بندی سیستم را ببینیم. دستورات متفاوتی برای انجام این کار وجود دارد و در این بخش با برخی از آنها آشنا می‌شویم.

مشاهده‌ی اطلاعات پارتیشن‌بندی با دستور lsblk

ساده‌ترین دستور برای مشاهده‌ی اطلاعات پارتیشن‌بندی، دستور lsblk می‌باشد. در جلسه‌ی قبل به طور مختصر با این دستور آشنا شدیم. بیایید در سیستمی که به تازگی نصب کردیم این دستور را امتحان کنیم:

```
[root@localhost ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sda	8:0	0	20G	0	disk	
├─sda1	8:1	0	953M	0	part	/boot
├─sda2	8:2	0	4.7G	0	part	/var
├─sda3	8:3	0	3.7G	0	part	/home
├─sda4	8:4	0	1K	0	part	
├─sda5	8:5	0	1.9G	0	part	[SWAP]
└─sda6	8:6	0	8.8G	0	part	/
sr0	11:0	1	942M	0	rom	

اگر به ستون NAME نگاه کنید، می‌بینید که به ما می‌گوید یک هارد دیسک با نام sda داریم که به ۶ قسمت تقسیم شده است. همانطور که می‌بینید در اینجا پارتیشن sda4 هم وجود دارد، اما همانطور که قبلاً گفتیم، sda4 یک پارتیشن Extended می‌باشد و به همین دلیل است که ظرفیتی برابر با ۱ کیلوبایت دارد؛ پس وجود sda4 در خروجی را با وجود یک پارتیشن قابل استفاده‌ی دیگر اشتباه نگیرید.

ستون MAJ:MIN، شماره‌ی دیوایس را مشخص می‌کند. کرنل لینوکس، به دیوایس‌های بلوکی و کاراکتری (جلسه‌ی قبل در مورد آنها صحبت کردیم) یک جفت شماره اختصاص می‌دهد. این شماره معمولاً با فرمت <major>:<minor> عنوان می‌شود، به طوری که <major> شماره‌ی اصلی و <minor> شماره‌ی فرعی می‌باشد. برخی از این شماره‌ها، برای دستگاه‌های خاص رزرو شده‌اند. شماره‌ی دیوایس مخصوص دستگاه‌های بلوکی SCSI، شماره‌ی ۸ می‌باشد. به همین دلیل در ستون MAJ:MIN، اولین عدد، ۸ می‌باشد. پس از این عدد، شماره‌ی فرعی را می‌بینید که از صفر شروع شده و به ازای هر پارتیشن، یک عدد به آن اضافه می‌شود. اگر یک دیوایس SCSI دیگر به سیستم اضافه کنیم، عدد اول آن، ۸ باقی می‌ماند اما عدد دوم آن متفاوت خواهد بود.

ستون RM، مشخص کننده‌ی removable بودن یا نبودن دستگاه می‌باشد. ستون SIZE نشان دهنده‌ی ظرفیت دیوایس مورد نظر می‌باشد. ستون RO، مشخص کننده‌ی Read Only بودن یا نبودن دستگاه می‌باشد. اگر در این ستون عدد ۱ مشاهده شود، یعنی دستگاه مورد نظر به صورت Read Only بر روی سیستم مانده شده است. ستون TYPE نشان دهنده‌ی نوع دستگاه می‌باشد. مثلاً sda، به اولین هارددیسک ما گفته می‌شود، به همین دلیل در ستون TYPE، مقدار disk برای آن وجود دارد؛ اما sda1 تا sda6، پارتیشن‌های sda می‌باشد،

به همین دلیل TYPE آنها part می‌باشد. یا مثلا دستگاه sr0 که اشاره به سی‌دی‌رام دارد، دارای TYPE برابر با rom می‌باشد.

در نهایت، ستون MOUNTPOINT نشان دهنده‌ی موقعیت مانیت هر دستگاه یا پارتیشن می‌باشد. توجه کنید که sda4 در جای خاصی مانیت نشده است، و باز هم می‌گوییم که دلیل این امر، Extended بودن این پارتیشن می‌باشد.

مشاهده‌ی اطلاعات پارتیشن‌بندی با استفاده از fdisk

fdisk برنامه‌ای است که می‌توان از آن برای پارتیشن‌بندی هارددیسک‌های سیستم استفاده کرد. ما در جلسه‌ی بعد به طور کامل در مورد این برنامه صحبت می‌کنیم، اما فعلا می‌خواهیم با استفاده از این دستور، اطلاعات پارتیشن‌بندی را ببینیم. برای دیدن پارتیشن‌های سیستم، کافی است این دستور را با آپشن -l اجرا کنیم:

```
[root@localhost ~]# fdisk -l
```

```
Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0003317a
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	1953791	975872	83	Linux
/dev/sda2		1953792	11718655	4882432	83	Linux
/dev/sda3		11718656	19529727	3905536	83	Linux
/dev/sda4		19529728	41943039	11206656	5	Extended
/dev/sda5		19531776	23531519	1999872	82	Linux swap / Solaris
/dev/sda6		23533568	41943039	9204736	83	Linux

همانطور که می‌بینید، با استفاده از آپشن -l، اطلاعات پارتیشن‌بندی برای هارددیسک sda، به ما داده شد. در ستون Device، نام دیوایس فایل مربوط به هر پارتیشن نوشته شده است. ستون Boot، مشخص می‌کند که کدام یک از پارتیشن‌ها، قابل بوت یا Bootable می‌باشند. ستون Start، نشان دهنده‌ی شماره‌ی سکتور شروع کننده‌ی یک پارتیشن می‌باشد و ستون End، نمایانگر شماره‌ی سکتور پایانی یک پارتیشن می‌باشد.

اگر به ستون Start دقت کنید، می‌بینید که اولین پارتیشن ما از سکتور ۲۰۴۸ شروع شده است. به نظر شما دلیل این امر چیست؟ همانطور که گفتیم، درایوهای ما به روش MBR پارتیشن‌بندی شده‌اند و در سیستم MBR، اولین سکتور مربوط به اطلاعات بوت‌لودر و جدول پارتیشن می‌باشد. اما چرا سایر این سکتورها خالی هستند؟ این سکتورها معمولا توسط بوت‌لودرهای پیشرفته نظیر GRUB2.0 به کار می‌روند. به عبارت دیگر این بخش توسط مرحله‌ی ۱،۵ بوت‌لودر استفاده می‌شود. بعدا که با بوت‌لودر GRUB آشنا شویم، این مفهوم را بهتر درک خواهید کرد.

ستون Block، نشان دهنده‌ی تعداد بلوک‌های موجود در هر پارتیشن می‌باشد. ابزار fdisk، هر بلوک را معادل ۱ کی‌بایت (معادل ۱،۰۲۴ کیلو‌بایت) قرار می‌دهد. یعنی مثلا پارتیشن اول ما، ۹۸۵۸۷۲ کی‌بایت معادل ۹۹۹۲۹۲،۹۲۸ کیلو‌بایت فضا دارد. توجه کنید که سایز هر بلوک در ابزارهای متفاوت، فرق خواهد داشت.

ستون Id، نشان دهنده‌ی Partition ID هر پارتیشن می‌باشد. این Id نشان دهنده‌ی فایل‌سیستم پارتیشن می‌باشد و علاوه بر این، می‌تواند نمایانگر وجود هر گونه flag خاص برای یک پارتیشن باشد. معنی هر Id را

در ستون System می‌توان مشاهده کرد. مثلاً 83، نشان دهنده‌ی پارتیشن معمولی سیستم‌های لینوکس می‌باشد. 82، نشان دهنده‌ی پارتیشن Swap می‌باشد و 5 نشان دهنده‌ی پارتیشن Extended می‌باشد.

مشاهده‌ی اطلاعات پارتیشن با استفاده از parted

همانطور که در بخش قبل دیدید، اطلاعات نمایش داده شده در مورد پارتیشن‌ها توسط fdisk، زیاد خوانا و مناسب نیست. parted، یک برنامه‌ی قوی برای پارتیشن‌بندی هارددیسک‌ها می‌باشد. ما در جلسه‌ی بعد به طور کامل با این برنامه نیز آشنا می‌شویم، اما فعلاً می‌خواهیم با استفاده از آن، اطلاعات پارتیشن‌بندی سیستم را مشاهده کنیم. برای این کار، کافی است این دستور را با آپشن -l اجرا کنیم:

```
[root@localhost ~]# parted -l
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

Number	Start	End	Size	Type	File system	Flags
1	1049kB	1000MB	999MB	primary	xfs	boot
2	1000MB	6000MB	5000MB	primary	xfs	
3	6000MB	9999MB	3999MB	primary	xfs	
4	9999MB	21.5GB	11.5GB	extended		
5	10.0GB	12.0GB	2048MB	logical	linux-swaps(v1)	
6	12.0GB	21.5GB	9426MB	logical	xfs	

همانطور که می‌بینید، خروجی این برنامه نیز بسیار شبیه به خروجی fdisk می‌باشد، با این تفاوت که خواناتر می‌باشد و اطلاعاتی نظیر سایز کلی هر پارتیشن و فایل سیستم آن نیز به شکل بهتری به ما نمایش داده می‌شود.

تحقیق: LVM چیست؟

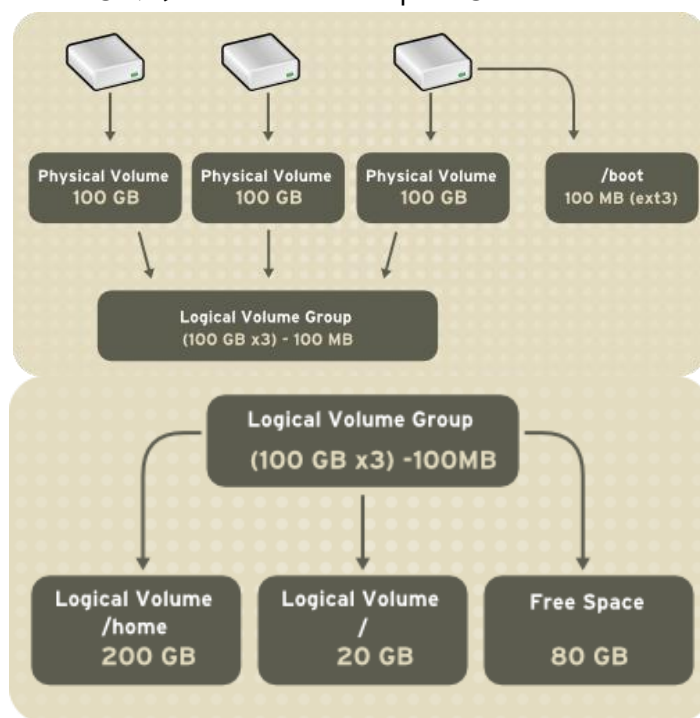
LVM یا Logical Volume Management، یک روش برای ساده کردن مدیریت هارددیسک‌ها می‌باشد. با کمک LVM، می‌توانیم به سادگی حجم یک پارتیشن را کم یا زیاد کنیم، یک هارددرایو را تعویض کنیم یا به راحتی به سیستم اضافه کنیم و همچنین قابلیت‌های پیشرفته‌تری نظیر snapshot گرفتن و mirror کردن هارددرایوها را داشته باشیم.

LVM ساختاری سه لایه‌ای دارد که این لایه‌ها، به شرح زیر می‌باشند:

- Physical Volumes
 - دستگاه‌های بلوکی هستند که زیربنای LVM می‌باشند. در واقع Physical Volume، صرفاً یک اسم فشنک برای دستگاه‌های ذخیره‌سازی معمولی (همان هارددیسک و SSD خودمان) می‌باشد.
- Volume Groups
 - در LVM، Physical Volume‌ها درون یک سری مخزن ذخیره‌سازی قرار می‌گیرند. در واقع Volume Group‌ها ویژگی‌های هر کدام از دستگاه‌های فیزیکی درون خود را از ما پنهان می‌کنند و به ما فقط یک دستگاه منطقی که فضایی برابر با مجموع کل فضای دستگاه‌های درونی آن می‌باشد، ارائه می‌دهند.
- Logical Volumes
 - هر Volume Group می‌تواند به چندین Logical Volume تقسیم شود. Logical Volume‌ها دقیقاً مثل پارتیشن‌ها در دیسک‌های فیزیکی می‌باشند، با این تفاوت که از انعطاف‌پذیری بیشتری برخوردار

هستند. کاربران هنگام کار با سیستم، معمولاً با Logical Volume ها کار می‌کنند.

پس به طور خلاصه، می‌توان گفت که ما از LVM برای ترکیب کلیه دستگاه‌های فیزیکی سیستم جهت جمع کردن کل فضای ذخیره‌سازی درون سیستم استفاده می‌کنیم و سپس می‌توانیم این فضای جمع شده را تبدیل به یک سری فضای کوچک‌تر کنیم که عملکردی نظیر پارتیشن‌های انعطاف‌پذیر خواهند داشت. انجام این کار، باعث می‌شود که بتوانیم به هر کدام از این «پارتیشن‌ها»، به صورت داینامیک، فضای بیشتر (فضایی که می‌تواند از چندین دیسک فیزیکی متفاوت باشد) اختصاص دهیم. بیایید با تعدادی تصویر این قضیه را بهتر درک کنیم:



تصویر ۱۵ - ساختار LVM

همانطور که در تصویر ۱۵ می‌بینید، در LVM، هر هارد دیسک، به چند Physical Volume تبدیل شده، سپس همه‌ی آن Physical Volume ها با هم ترکیب شده و یک Volume Group تشکیل می‌دهند. سپس ما می‌توانیم این Volume Group ها را تقسیم به چند Logical Volume یا پارتیشن کنیم که سپس آنها را می‌توانیم در Mount Point های متفاوت قرار دهیم.