

(( به نام خدا ))

پارتیشن چیست؟

پارتیشن ابزاری برای تقسیم یک هارد دیسک به چند درایو منطقی است. یک پارتیشن مجموعه‌ای از بلوک‌های پشت سر هم در یک درایو است که با آن به صورت یک دیسک مستقل رفتار می‌شود. یک جدول پارتیشن (partition table) شاخصی است که بخش‌های هارد دیسک را به پارتیشن‌ها مرتبط می‌کند.

چرا چند پارتیشن داشته باشیم؟

- کپسوله کردن اطلاعات. در صورت آسیب دیدن فایل سیستم، یک پارتیشن آسیب می‌بیند و در اثر وقوع حادثه شما فقط بخشی از اطلاعاتتان را از دست می‌دهید.
- افزایش کارایی فضای خالی دیسک. شما بر اساس نیازتان پارتیشن‌ها را با اندازه بلوک‌های مختلفی فرمت می‌کنید. اگر داده‌های شما در تعداد زیادی فایل کوچک (کمتر از یک کیلوبایت) باشد و شما از اندازه بلوک‌های 4 کیلوبایتی استفاده کنید، به ازای ذخیره سازی هر فایل، 3 کیلوبایت فضا را هدر می‌دهید. معمولاً شما برای هر فایل، نیمی از اندازه‌ی هر بلوک را هدر می‌دهید، بنابراین اگر تعداد زیادی فایل در اختیار دارید، انتخاب کردن اندازه بلوک به اندازه‌ی متوسط حجم فایل‌هایتان، مهم است.
- محدود کردن رشد داده‌ها. پروسه‌های فراری یا کاربران دیوانه می‌توانند فضای دیسک زیادی از سیستم عامل بگیرند به نحوی که سیستم عامل حتی برای سازماندهی خودش هم فضایی در اختیار نداشته باشد. این می‌تواند منجر به فاجعه شود. با جدا سازی فضا، مطمئن می‌شوید که پروسه‌هایی غیر از سیستم عامل در اثر کافی نبودن فضای خالی، از بین خواهند رفت.

محدودیت‌ها

- پارتیشن‌ها نباید همپوشانی (فضای مشترک) داشته باشند. این مسئله، باعث آسیب دیدن اطلاعات و اتفاقات غیر قابل پیش بینی می‌شود.
- نباید بین پارتیشن‌های همجوار، فضای خالی وجود داشته باشد. با وجود مضر نبودن این مسئله، با قرار دادن فضای خالی بین پارتیشن‌ها، بخشی از فضای خالی دیسک را هدر می‌دهید.
- دیسک را می‌توان به طور کامل، پارتیشن بندی نکرد. شما ممکن است تصمیم بگیرید مقداری فضای خالی پارتیشن نشده در انتهای دیسکتان داشته باشید.
- پارتیشن‌ها را نمی‌توان انتقال داد، اما آن‌ها را می‌توان تغییر اندازه داد یا کپی کرد. (fdisk این موضوع را پشتیبانی نمی‌کند!)

چند برنامه پارتیشن بندی :

اسم برنامه	توضیحات
sfdisk	یک نسخه از برنامه Fdisk برای محیط ترمینال
cfdisk	یک نسخه از برنامه Fdisk با پشتیبانی اشاره‌گر
parted	ویرایشگر پارتیشن گنو
Partition Magic	یک برنامه‌ی سودمند تجاری، برای ساختن، تغییر اندازه، تلفیق و تبدیل کردن پارتیشن‌ها بدون از دست رفتن اطلاعات
Disk Drake	یک برنامه بر پایه‌ی perl/Gtk برای ساختن، تغییر اندازه و حذف کردن پارتیشن‌ها

ابزارها:

مجموعه اصطلاحاتی که کاربران لینوکس استفاده می‌کنند، باید دانسته شود. در لینوکس، پارتیشن‌ها با فایل‌های ابزار ارائه می‌شوند. این فایل‌های غلط انداز در /dev/ قرار دارند، خروجی زیر را ببینید:

```
brw-rw---- 1 root disk 3, 0 May 5 1998 hda
brw-rw---- 1 root disk 8, 0 May 5 1998 sda
```

یک فایل ابزار، یا با c شروع می شود (ابزارهای "character"، ابزارهایی هستند که از کش بافر استفاده نمی کنند) یا b (برای ابزارهای "block" که از کش بافر عبور داده می شوند) در لینوکس همه ی دیسک ها به عنوان ابزارهای block ارائه می شوند.

اسامی ابزارها:

بر اساس قرارداد، به درایو های IDE اسامی /dev/hda تا /dev/hdd داده می شود. دیسک اول a، دیسک دوم b و الی آخر. به عنوان مثال، /dev/hda اولین درایو روی اولین کنترلر IDE می باشد و /dev/hdd دومین دیسک روی دومین کنترلر IDE است (چهارمین دیسک IDE در کامپیوتر). شما می توانید بر روی این دیسک ها با استفاده از car یا dd مستقیماً بنویسید، البته چون این فایل ابزارها تمام دیسک را ارائه می کنند، با شروع کردن از اولین بلوک، شما می توانید به اشتباه روی رکورد اصلی بوت Master Boot Record و جدول پارتیشن ها partition table نوشته و کل دیسک را غیر قابل استفاده کنید.

با پارتیشن کردن یک دیسک ها، پارتیشن ها به صورت شماره هایی که به انتهای اسامی دیسک ها اضافه می شوند، ارائه می شوند. به عنوان مثال، دومین پارتیشن روی دومین دیسک با مسیر و عنوان /dev/hdb2 مشخص خواهد شد. دیسک های اسکازی نیز از الگوی مشابهی استفاده می کنند. با این تفاوت که آن ها از sd به جای hd استفاده می کنند. اولین پارتیشن از دومین دیسک اسکازی چنین خواهد بود:

/dev/sdb1

پارتیشن های اولیه (primary) دیسک شماره های 1، 2، 3 و 4 هستند و پارتیشن های منطقی از 5 شروع می شوند.

انتساب نام به دیسک های اسکازی:

در سان (solaris) و آیریکس (SGI) به درایو های اسکازی اسمی بر اساس محل اتصالشان به سیستم داده می شود. ولی در لینوکس، دیسک با SCSI ID کمتر، حرف ابتدایی تری از الفبای انگلیسی را می گیرد، بنابراین اگر یکی از درایوها را از زنجیره خارج کنید، اسم دیسک با شماره SCSI ID بیشتر تغییر خواهد کرد. اگر شما بر روی لینوکستان دو کنترلر اسکازی داشته باشید، برای تشخیص اسم دیسک، باید خروجی /bin/dmmsg را ببینید. اگر یکی از دو کنترلر ها را جدا کنید، بقیه ی دیسک های متصل به کنترلر باقی مانده، تغییر نام خواهند داد.

شماره ی ابزارها:

مهمترین مسئله ی مربوط به یک ابزار، شماره ی Major و Minor ابزار هاست که به جای اندازه ی فایل نمایش داده می شوند.

\$ ls -l /dev/hda

brw-rw----	1	root	disk	3	0	May 5 1998	hda
اجازه ها		مالک	گروه	شماره ی مژور ابزار	شماره مینور دستگاه	تاریخ	اسم ابزار

جدول ۲. ویژگی های فایل های ابزار

وقتی به یک فایل ابزار دسترسی پیدا می کنید، کد Major مشخص می کند که کدام درایور ابزار، برای اجرای عملیات ورودی/خروجی فراخوانی شوند. این فراخوانی با فرستادن کد Minor به عنوان پارامتر درایور انجام می شود و کاملاً بسته به درایور تفسیر متفاوتی از آن می شود. معمولاً مستندات درایور توضیح می دهند که چطور درایور از شماره ی Minor استفاده می کند. برای دیسک های IDE کدهای Major معمولاً 3,22,33,34 هستند و برای دیسک های اسکازی کد Major معمولاً 8 است. کدهای Major و Minor هر کدام یک بایت هستند به همین دلیل تعداد پارتیشن های هر دیسک محدود است.

انواع پارتیشن ها:

یک پارتیشن برای میزبانی فایل سیستم، برچسب می خورد. فایل سیستم می تواند، استاندارد لینوکس مثل فایل سیستم ext2 یا swap یا حتی یک فایل سیستم خارجی مثل (Microsoft (NTFS یا (SUN (UFS باشد. برچسب، یک کد عددی است که به

پارتیشن (صرفنظر از نوع آن) متصل می‌شود. به عنوان مثال، برچسب ext2 کد 0x83 است و Linux Swap کد 0x82 را دارد.

پارتیشن اولیه:

تعداد پارتیشن‌های ممکن روی سیستم‌های مبتنی بر Intel از آغاز محدود بوده است. جدول پارتیشن اصلی (partition table)، که به عنوان بخشی از سکتور بوت نصب شده است، فقط برای ورودی چهار پارتیشن فضا دارد. این پارتیشن‌ها به عنوان پارتیشن اولیه (primary) شناخته می‌شوند.

پارتیشن منطقی:

یک پارتیشن اولیه‌ی دیسک، می‌تواند به چند زیر پارتیشن تقسیم شود. به هر یک از این زیر پارتیشن‌ها، پارتیشن منطقی می‌گویند. این راهکار به طور موثری به ما اجازه می‌دهد که از حصار چهار پارتیشن عبور کنیم.

از پارتیشن اولیه، که برای میزبانی پارتیشن‌های منطقی استفاده می‌شود به عنوان یک پارتیشن توسعه یافته Extended یاد می‌شود که فایل سیستم خاص خودش (0x05) را دارد. برخلاف پارتیشن‌های اولیه، پارتیشن‌های منطقی حتماً می‌بایست پشت سر هم باشند. هر پارتیشن منطقی، شامل اشاره‌گری به پارتیشن منطقی بعدی است که نشان می‌دهد که تعداد پارتیشن‌های منطقی، نامحدود است. البته خود لینوکس محدودیتی به مجموع تعداد پارتیشن‌ها اعمال می‌کند. در نتیجه یک محدودیت تعداد برای پارتیشن‌های منطقی وجود دارد. در کل، برای دیسک‌های آسکازی حداکثر 15 درایو منطقی و برای دیسک‌های IDE حداکثر 63 درایو منطقی می‌تواند وجود داشته باشد.

پارتیشن‌های SWAP ::

به هر پروسه‌ای که بر روی کامپیوتر شما اجرا می‌شود تعدادی بلوک حافظه‌ی RAM تخصیص می‌یابد. به این بلوک‌ها، page می‌گویند. به مجموعه‌ی page‌های حافظه که در آینده‌ی نزدیک توسط CPU مورد مراجعه قرار می‌گیرند، working set گفته می‌شود. لینوکس تلاش می‌کند تا این دسترسی‌ها را (با تصور اینکه page‌های تازه استفاده شده در آینده‌ی نزدیک نیز مجدداً مورد استفاده قرار می‌گیرند) پیش بینی کند و در صورت امکان، این page‌ها را در حافظه‌ی RAM نگه دارد.

اگر شما، پروسه‌های بسیار زیادی بر روی یک ماشین داشته باشید، کرنل تلاش خواهد کرد تا مقداری از RAM را بانوشتن page‌ها بر روی دیسک خالی کند. این مسئله، دلیل وجود فضای swap است. این فناوری، مقدار حافظه‌ی در دسترس را به طرز موثری افزایش می‌دهد. البته، عملیات ورودی/خروجی دیسک در برابر عملیات خواندن/نوشتن حافظه‌ی RAM خیلی آهسته‌تر است.

اگر حافظه بسیار کم شود، کرنل page‌ها را از working set یک پروسه خارج می‌کند تا page‌های پروسه‌ی دیگر را وارد کند، در این وضعیت گفته می‌شود که ماشین در حال thrashing است. (این در واقع وقتی است که زمان سیستم بیشتر برای page کردن مصرف می‌شود تا اجرای برنامه‌ها) که در این حالت، کارایی سیستم با نرخ تقریبی سرعت دسترسی حافظه نسبت به سرعت دسترسی دیسک کاهش پیدا می‌کند. فضای swap چیزی است که شما به آن نیاز خواهید داشت، اما جایگزینی برای مقدار RAM کافی نیست.

چند نتیجه از بحث‌ها:

پارتیشن بوت:

بهرتر است که پارتیشن بوت یک پارتیشن اولیه باشد نه یک پارتیشن منطقی. این مسئله فرآیند بازیابی را در مواقع بروز فاجعه آسان خواهد کرد، اما این موضوع از نظر تکنیکی الزامی نیست. پارتیشن بوت، باید از نوع "linux native" 0x83 باشد. اگر از lilo استفاده می‌کنید، پارتیشن بوت شما، باید اولین 1024 سیلندر دیسک را شامل شود. (معمولاً پارتیشن بوت، فقط شامل تصویر کرنل می‌باشد).

اگر بیش از یک پارتیشن بوت دارید (از سیستم عامل‌های دیگر، به عنوان مثال) همه‌ی آن‌ها را در اولین 1024 سیلندر نگه دارید (تمام پارتیشن‌های DOS باید در اولین 1024 سیلندر باشند) ولی اگر برای بارگذاری کرنل از ابزاری غیر از lilo استفاده می‌کنید (به عنوان مثال، از یک دیسک بوت یا از LOADLIN.EXE که یک بارگذارنده‌ی لینوکس (linux loader) برپایه SMS-DOS است) پارتیشن می‌تواند هر کجا باشد.

پارتیشن swap

برای استفاده از ویژگی swap شما به یک پارتیشن swap نیاز خواهید داشت. این پارتیشن حتماً باید از نوع "Linux" 0x82 swap باشد و آن را می‌توان در هر محدوده‌ای از دیسک در نظر گرفت (اما برخی نکات در در نظر بگیرید 442). از هر دو نوع پارتیشن primary یا logical می‌توان برای فضای swap استفاده کرد. می‌توان بر روی یک دیسک بیش از یک پارتیشن swap داشت. در کل، بر روی یک سیستم (مستقل از تعداد دیسک‌ها) می‌توان حداکثر 8 پارتیشن swap اختیار کرد.

پارتیشن logical (منطقی)

حتماً باید از یک پارتیشن primary به عنوان نگهدارنده (همان پارتیشن extended) برای پارتیشن‌های منطقی استفاده شود. پارتیشن extended، می‌تواند در هر محدوده‌ای از دیسک قرار داشته باشد. پارتیشن‌های منطقی باید پشت سر هم باشند، اما

الزاما نیازی نیست تا تمام فضای پارتیشن extended را پر کنند.

فایل سیستم‌ها:

کدام فایل سیستم‌ها به پارتیشن خودشان نیاز دارند؟

هر چیزی در فایل سیستم لینوکس، می‌تواند در یک پارتیشن مستقل قرار بگیرد. البته، شرایطی وجود دارد که شما ممکن است بخواهید از رشد فایل سیستم خاصی جلوگیری کنید. به عنوان مثال، اگر فضای ایمیل شما، در یک محل مشترک با فضای فایل سیستم ریشه (/) باشد، و این فضا، باقی مانده‌ی فضای پارتیشن را پر کند، کامپیوتر شما اساسا هنگ خواهد کرد.

/var

این فایل سیستم، شامل پوشه‌های مربوط به دستگاه‌های جانبی درون خطی مثل ایمیل و عملیات چاپ است. همچنین، این فایل سیستم گزارشات خطای سیستم را نگه می‌دارد. اگر سیستم شما، یک کامپیوتر سرور است که پیغام‌های متعددی تولید می‌کند، آن پیغام‌ها می‌توانند فضای پارتیشن را پر کنند. کامپیوترهای سرور باید /var را در پارتیشنی غیر از / داشته باشند.

/usr

این آنجایی است که بیشتر فایل‌های اجرایی باینری به آنجا می‌روند. همچنین، شاخه‌ی سورس کرنل به همراه بسیاری مستندات اینجاست.

/tmp

بعضی برنامه‌ها، فایل‌های داده‌ای موقت خود را در اینجا می‌نویسند که معمولا، فایل‌هایی بسیار کوچک هستند. البته، اگر شما وظیفه‌های سنگین پردازشی مثل برنامه‌های علمی یا نرم افزارهای مهندسی، اجرا کنید به صدها مگابایت فضا فقط برای اجرای بخشی از عملیات نیاز خواهید داشت. در این صورت، باید /tmp را در پارتیشنی غیر از / نگه دارید.

/home

اینجا پوشه‌های خانگی کاربران قرار دارند. اگر شما اجباری به استفاده از سهمیه بندی quota برای کاربران ندارید، این فایل سیستم می‌تواند در پارتیشن خودش باشد.

/boot

این فایل سیستم، محل قرار گرفتن، image‌های هسته‌ی لینوکس (kernel) است. اگر از MS DOS استفاده می‌کنید، این فایل سیستم باید در اولین 1024 سیلندر دیسک قرار بگیرد. وظیفه‌ی شما این است که اطمینان حاصل کنید که پارتیشن مربوط به این فایل سیستم در آن موقعیت قرار می‌گیرد تا lilo بتواند آن را ببیند. اگر دیسکی بزرگ‌تر از 1024 سیلندر در اختیار دارید، گذاشتن این فایل سیستم در اولین پارتیشن دیسک، دیده شدنش توسط lilo را تضمین می‌کند.

طول عمر فایل و دوره‌ی زمانی پشتیبان‌گیری و نقش آن‌ها در پارتیشن بندی:

برای ext2، تصمیمات پارتیشن بندی باید با ملاحظات پشتیبان‌گیری و برای جلوگیری از تفرقه‌ی خارجی external fragmentation ناشی از فایل‌هایی با طول عمر متفاوت اجرا شود.

فایل‌ها طول عمر دارند. پس از اینکه فایل‌ی ساخته می‌شود، مدتی بر روی سیستم می‌ماند و پس از آن حذف خواهد شد. طول عمر فایل بسته به سیستم و گاه بسته به مسیر فایل بسیار متفاوت است. به عنوان مثال، فایل‌های موجود در /bin یا /sbin یا /usr/bin یا /usr/sbin و دایرکتوری‌های مشابه تقریبا عمری بسیار طولانی دارند: چندین ماه و بیشتر. فایل‌های در /home تقریبا طول عمر متوسطی در حدود چندین هفته یا همین حوالی دارند. فایل‌های /var معمولا حیات کوتاهی دارند، اغلب هیچ فایل‌ی در مسیر /var/spool/news بیشتر از چند روز باقی نمی‌ماند، فایل‌های در مسیر /var/spool/lpd نیز عمری در حدود چند دقیقه یا کمتر دارند.

برای پشتیبان‌گیری، بهتر است حجم پشتیبان‌های روزانه کمتر از حجم یک رسانه‌ی خام مورد استفاده برای پشتیبان‌گیری باشد. پشتیبان‌گیری روزانه می‌تواند یک پشتیبان‌گیری کامل، یا یک پشتیبان‌گیری مقطعی باشد.

شما می‌توانید تصمیم بگیرید که حجم پارتیشن به قدری کوچک باشد که به طور کامل روی رسانه‌ی ذخیره کننده‌ی پشتیبان‌ها جا بگیرد (با انتخاب پشتیبان‌گیری کامل روزانه). در هر صورت یک پارتیشن باید به اندازه‌ای کوچک باشد که تفاوت‌های روزانه (تمام فایل‌های تغییر یافته) بر روی یک رسانه‌ی خام ذخیره کننده‌ی پشتیبان‌ها جا بگیرد (با انتخاب پشتیبان‌گیری مقطعی و انتظار تغییر رسانه به صورت هفتگی/ماهانه به صورت کامل - در این حالت عملیات پشتیبان‌گیری بدون نظارت ممکن نیست)

سیاست پشتیبان‌گیری به تصمیم‌گیری در مورد اندازه‌ی پارتیشن‌ها، بستگی دارد.

به یاد داشته باشید که در طرح ریزی و خرید فضای خالی دیسک، بودجه ی کافی برای پشتیبان گیری کنار گذاشته باشید! اطلاعات پشتیبان گیری نشده بی ارزش است! هزینه های تولید مجدد اطلاعات بسیار بیشتر از هزینه های پشتیبان گیری است.

برای افزایش کارایی بهتر است که فایل های با طول عمر متفاوت را در پارتیشن های متفاوت بگذارید. با این روش، فایل های کم عمر روی پارتیشن اخبار ممکن است به طور سنگینی دچار تفرقه شوند، در حالیکه فشاری بر روی کارایی پارتیشن / home یا /وارد نخواهد شد.

#### 4.4 پارتیشن swap

اگر تصمیم گرفته اید که یک پارتیشن را به swap اختصاص دهید که در کل تصمیم خوبی است، از اصول زیر برای تخمین فضای آن استفاده کنید.

- در لینوکس فضای RAM و swap با هم جمع می شوند (این مسئله برای همه ی لینوکس ها صادق نیست). به عنوان مثال، اگر شما 8 مگابایت RAM و 12 مگابایت فضای swap داشته باشید، در کل حدود 20 مگابایت حافظه ی مجازی دارید.
- وقتی فضای swap را تنظیم می کنید، باید حداقل 16 مگابایت حافظه ی مجازی داشته باشید. بنابراین برای 4 مگابایت RAM باید حداقل 12 مگابایت فضای swap در نظر بگیرید و برای 8 مگابایت RAM باید حداقل 8 مگابایت فضای swap بسازید.
- در حال حاضر حداکثر فضای پارتیشن swap وابسته به معماری کامپیوتر است. برای دانستن حداکثر فضای swap به جدول زیر مراجعه کنید.

معماری	حداکثر حجم swap قابل پشتیبانی
sparc	1Gb
i386 & PowerPC	2Gb
alpha	128Gb
sparc64	3Tb

برای کرنل های نسخه ی 2.1 و قبل از آن، حجم فضای swap محدود به 128mb است. پارتیشن می تواند فضایی بیشتر از 128 مگابایت داشته باشد اما فضای خالی اضافه، هرگز استفاده نخواهد شد. اگر به بیش از 128 مگابایت فضا بر روی یک کرنل 2.1 نیاز داشته باشید، باید چند پارتیشن swap بسازید. برای جزییات بیشتر صفحه ی راهنمای man مربوط به mkswap را بخوانید.

در تخصیص فضای swap به یاد داشته باشید که فضای swap بیش از حد احتمالا اصلا مفید نیست. یک قانون خیلی قدیمی سرانگشتی در روزهای PDP و Vax این بود که اندازه ی working set یک برنامه تقریبا 25% حجم مجازی اش است. بنابراین احتمالا اختصاص دادن فضایی بیشتر از سه برابر حافظه ی RAM بلا استفاده خواهد بود.

اما به یاد داشته باشید که این فقط یک قانون سرانگشتی است. به راحتی می توان سناریوهایی ساخت که برنامه ها working set هایی بسیار بزرگ یا بسیار کوچک داشته باشند. به عنوان مثال، یک برنامه ی شبیه سازی با مجموعه داده های بزرگ که با سبکی خیلی تصادفی دسترسی شود تقریبا هیچ موقعیت قابل بیانی برای مراجعه ی سگمنت های داده اش نخواهد داشت، بنابراین working set اش بسیار بزرگ خواهد بود.

از طرف دیگر، یک برنامه ی xv (یک ویرایشگر تصویر قدیمی) با تعداد زیادی تصویر JPEG باز شده به صورت همزمان، که همه بجز یکی آیکنیفای (کوچک بودن پنجره به نحوی که فقط آیکنش مشخص باشد) شده اند، سگمنت داده ای بسیار بزرگ خواهد داشت. اما با پایان یافتن ویرایش یک تصویر، بیشتر حافظه ی مربوط به xy دست نمی خورد. این حالت برای یک ویرایشگر متنی با پنجره های ویرایش بسیار زیاد که فقط یک پنجره ی آن در یک زمان ویرایش می شود، صدق می کند. این برنامه ها- اگر به درستی طراحی شده باشند-، موقعیت مکانی بزرگی برای مراجعه دارند و بخش های زیادی از آن ها می توانند بدون ایجاد فشار زیادی به روی کارایی سیستم swap شوند.

ممکن است کسی مشکوک شود که عدد 25% مربوط به زمان خط فرمان، دیگر در دوره ی برنامه های جدید تحت رابط گرافیکی که چندین سند را ویرایش می کنند، صدق نمی کند. اما تا جایی که می دانم، فعلا هیچ سند معتبر جدیدی برای تایید این ارقام نیامده است.

بنابراین، برای یک سیستم با پیکربندی 16 مگابایت RAM، نیازی به فضای swap در پایین ترین پیکربندی ندارد و برای این سیستم بیش از 48 مگابایت فضای swap احتمالا بلا استفاده خواهد بود. مقدار دقیق حافظه ی مورد نیاز به ترکیب برنامه های ماشین بستگی دارد. ( چه انتظاری داشتید؟)

فضای swap را کجا باید در نظر بگیریم؟

المان‌های مکانیکی آهسته هستند، المان‌های الکترونیکی سریع هستند. هارد دیسک‌های جدید، هدهای زیادی دارند. سوییچ کردن بین هدهای تراک مشابه سریع است، به خاطر اینکه کاملاً الکترونیکی است. سوییچ کردن بین تراک‌ها کند است، به خاطر اینکه به یک حرکت هِد در دنیای واقعی نیاز دارد.

بنابراین، اگر دیسکی با هدهای زیاد و یک دیسک با تعداد کمی هِد دارید و هر دو در سایر پارامترهای یکسان هستند، دیسک با هدهای بیشتر سریعتر است.

تقسیم کردن فضای swap و قرار دادنش بین دو دیسک حتی باز هم سریعتر خواهد بود.

دیسک‌های قدیمی تعداد یکسانی سکتور در تمام تراک(شیار)ها دارند. با این دیسک‌ها، قرار دادن فضای swap در میانه‌ی دیسک‌ها، با در نظر گرفتن اینکه هدهای دیسک از یک تراک تصادفی به سمت فضای swap حرکت می‌کنند، سریعترین روش خواهد بود.

دیسک‌های جدیدتر، از ZBR (محدوده‌ی ذخیره‌ی بیت) استفاده می‌کنند. آن‌ها سکتورهای بیشتری روی تراک (شیار)های خارجی دارند، با یک سرعت ثابت rpm (دور بر دقیقه). پس واضح است که کارایی در تراک(شیار)های خارجی بسیار بیشتر از تراک‌های داخلی است. فضای swap را روی تراک‌های سریع قرار دهید.

البته، هدهای دیسک شما تصادفی حرکت نمی‌کنند. اگر شما فضای swap را در میانه‌ی یک دیسک بین یک پارتیشن home پر استفاده و یک پارتیشن archive که اغلب بلا استفاده است، داشته باشید، بهتر است تا اینکه برای کمتر کردن جابجایی هِد، فضای swap را وسط پارتیشن home گذاشته باشید. حتی بهتر خواهد بود اگر فضای swap را بر روی دیسک بلا استفاده‌ی دیگری بگذارید.

خلاصه:

swapتان را روی یک دیسک سریع با هدهای زیاد که به انجام کار دیگری مشغول نیست بگذارید. اگر چند دیسک در اختیار دارید، swap را چند قسمت کنید و بین همه‌ی دیسک‌ها تقسیمش کنید یا حتی آن را روی دیسکی که به کنترلگر دیسک دیگری متصل است، بگذارید.

حتی از این هم بهتر اینکه، RAM بیشتری بخريد.

با آرزوی موفقیت جامعه‌ی متن باز ایران

نویسنده: علی قنوتیان  
نوع انتشار: آزمایشی (در حال ویرایش)  
مجوز: GNU/FDL

گروه کاربران لینوکس خوزستان

