آریا خلیق

تمرین شماره ۷ OS!

اواخر بهار ۹۸

۱)

کار DMA: پردازنده دستور I/O را به DMA می‌دهد و خود مشغول کار دیگری می‌شود. زمانی که DMA آن دستور I/O را انجام داد از طریق یک وقفه پردازنده را مطلع می‌کند.

DMA مکانیزم‌های متفاوتی دارد. در حالت اول تمامی ماژول‌ها از یک bus مشترک استفاده می‌کنند. DMA به عنوان یک پردازنده جایگزین داده را بین حافظه و I/O ماژول جابه‌جا می‌کند. (از programmed I/O استفاده می‌شود.) این روش بسیار زمان‌بر است و ارزان است زیرا به خاطر کنترل پردازنده روی programmed I/O انتقال هر word زمان دو باس را آشغال می‌کند (یک بار برای ارسال درخواست انتقال و بار دیگر برای خود انتقال)

روش دوم:

در این حالت یک راه بین DMA و یک یا چند ماژول I/O وجود دارد که متصل به bus نیستند. در این حالت منطق DMA می‌تواند یک بخش از خود ماژول I/O باشد یا در یک ماژول باشد که یک یا چند ماژول I/O را کنترل کند.

در روش بعد این ایده را پیشرفت می‌دهیم و از یک I/O BUS برای اتصال ماژول‌های I/O به DMA استفاده می‌کنیم. این‌کار باعث کاهش اینترفیس‌های داخل DMA به تنها یک عدد و تنظیمات قابل گسترش و تغییر می‌شود.

در مورد BUS:

در روش اول انتقال اطلاعات و موارد دیگر بین DMA و ماژول‌های I/O از طریق BUS اصلی انجام می‌شود.

در هر دو روش ۲ و ۳، DMA از BUS سیستم تنها برای انتقال داده و سیگنال‌های کنترلی استفاده می‌کند و کارهایش با I/O ماژول‌ها به صورت جداگانه و نه از روی این BUS انجام می‌شود.

۲)

دیسک‌ها ارزان‌تر شده‌اند و هر کامپیوتر می‌تواند چند دیسک داشته باشد. اگر دیسک‌ها به صورت موازی کار کنند نرخ انتقال داده می‌تواند خیلی بالا برود. به علاوه یک داده را می‌توان برای افزایش ضریب اطمینان از حفظ آن روی چندین دیسک ذخیره کرد که در صورت مشکل یکی داده هنوز هم در دیگری در دسترس باشد. این تکنیک‌ها RAID نامیده می‌شوند.

در گذشته RAIDها که از چندین دیسک تشکیل شده بودند راه‌حلی به صرفه برای یک دیسک با حجم زیاد بودند ولی اکنون RAIDها برای اطمینان و نرخ انتقال بیشتر استفاده می‌شوند.

RAID می‌تواند روی سیستم، یا سخت‌افزار خود دیسک‌ها پیاده شود.

سطوح مختلف:

برای رسیدن به تعادل بین هزینه و عمل‌کرد و اطمینان شماهای مختلفی برای RAID وجود دارد که هر کدام از این‌ها را به سطوح مختلف RAID نام‌گذاری کرده‌اند.

سطح ۰:

در این حالت برای هیچ داده‌ای تکرار نداریم و همه دیسک‌ها به صورت هم‌زمان از داده پر می‌شوند(data striping).

سطح ۱:

تمامی دیسک‌ها mirror می‌شوند و داده‌های یک دیسک در دیسک دیگر تکرار می‌شود.

سطح ۲:

در دیسک‌های دیگری parity یک داده ذخیره می‌شود که باعث می‌شود حجم کم‌تری برای این داده‌های redundancy مصرف کنیم. در این حالت اگر بیتی دچار مشکل شد از طریق memory system شناسایی می‌شود و می‌توان برطرف کرد.

سطح ۳:

بهبود یافته سطح ۲ است و در این حالت disk controllerها می‌توانند تشخیص دهند که یک sector درست خوانده شده است یا خیر و parity bit می‌تواند هم برای تشخیص و هم برای رفع خطا استفاده شود.

سطح ۴:

مانند سطح ۰ از انتقال هم‌زمان و تکراری داده‌ها در سطح بلوک استفاده می‌کند و یک parity-block را در یک دیسک دیگر برای خطا نگه‌داری می‌کند. اگر یک block مشکل داشت از طریق parity متوجه شده و آن را از روی دیسک دیگر بازیابی می‌کنیم.

سطح ۵:

مانند سطح ۴ است ولی داده‌ها و parityها را بین N+1 (تمام) دیسک‌ها پخش می‌کند(در سطح ۴ N تا برای داده و یکی برای parity بود). برای هر بلاک یک دیسک بلاک را نگه می‌دارد و دیسک دیگر parity را نگه می‌دارد.

سطح ۶:

مانند سطح ۵ است ولی داده تکراری بیشتری نسبت به آن برای محافظت در برابر از دست رفتن داده، ذخیره می‌کند. به جای parity هم از کدهای بازیابی خطای(همان بازیابی داده‌ای که خطا دارد خودمان!) Reed-Solomin استفاده می‌کند.

سطح ۱+۰:

برای ترکیب سطح ۱ و ۰ استفاده می‌شود. سطح ۰ برای بازدهی بسیار عالی است و سطح ۱ برای اطمینان بیشتر. این حالت در جاهایی استفاده می‌شود که بازدهی و اطمینان هر دو مهم هستند.

۳)

همه می‌دانیم که طبق آمار locality of reference داریم. Cache که حافظه کوچک و سریعی است این داده‌های محدود را در خود ذخیره می‌کند و از این طریق باعث کاهش زمان دسترسی به حافظه می‌شود. دقیقاً همین ایده را می‌توان برای دیسک استفاده کرد. کش یک دیسک یک بافر در حافظه اصلی است که sectorها را در خود ذخیره می‌کند.

در این حالت sectorهایی که قبلاً به آن دسترسی پیدا کرده‌ایم در داخل cache ذخیره می‌شوند و در صورتی که بخواهیم دوباره به آن دسترسی داشته باشیم از cache می‌خوانیم.

برای بازدهی بالا می‌خواهیم به یک miss ratio مشخص و کم برسیم. این بستگی به رفتار locality درخواست داده‌ها، الگوریتم جایگزینی در cache و فاکتورهای طراحی دارد.

کش همچنین می‌تواند زمان انتظار برای نوشتن روی دیسک را کاهش دهد.

۴)

روش ساده و پربازده FAT در سیستم‌عامل MS-DOS استفاده می‌شد. یک section در شروع هر volume برای نگه‌داشتن جدول، انتخاب و کنار گذاشته می‌شد. این جدول یک سطر برای هر بلاک دیسک دارد و به وسیله شماره بلاک ایندکس می‌شود. FAT شبیه linked list است. ورودی شماره اولین بلاک از فایل را دارد. آن سطر جدولی که مربوط به آن شماره بلاک است، شماره بلاک بعدی آن فایل را دارد. این زنجیره تا رسیدن به آخرین بلاک فایل ادامه پیدا می‌کند. زمانی که به EOF رسید (در جدول) دیگر تمام می‌شود. یک بلاک استفاده نشده به وسیله ۰ در جدول نمایش داده می‌شود. دادن یک بلاک خالی به یک فایل به سادگی پیدا کردن ۰ در این جدول است و زمانی که پیدا شد EOF قبلی با آدرس بلاک جدید جایگزین می‌شود.

استفاده از FAT باعث افزایش جابجایی head می‌شود(در صورتی که FAT کش نشود). زیرا head باید به اول volume برای خواندن آدرس بلاک برود و دوباره برای خواندن بلاک جابه‌جا شود. زمان random-access بهبود می‌یابد زیرا head می‌تواند آدرس هر بلاک را بسیار ساده و از طریق خواندن FAT پیدا کند.

۵)
توسعه‌دهنده‌های ویندوز یک فایل‌سیستم جدید به نام NTFS برای رفع نیازهای سرورها و workstationها توسعه داده‌اند.
NTFS یک فایل‌سیستم قدرت‌مند و منعطف است که روی فایل سیستم ساده ساخته شده است.

ویژگی‌های آن عبارتند از:
قابلیت بازیابی: یکی از مهم‌ترین ویژگی‌ها است و قابلیت بازیابی از crash کردن سیستم و دیسک دارد. در زمان‌هایی از این قبیل NTFS می‌تواند volumeهای دیسک را از نو بازسازی کند و آن را به حالت پایدار برگرداند. برای این کار از روش transaction-processing model استفاده می‌کند که هر تغییر مهمی به صورت اتمیک اجرا می‌شود، یعنی یا انجام می‌شود یا نمی‌شود. یعنی در زمانی که crash و ... اتفاق افتاد یا تغییر انجام می‌شود یا به حالت قبل برمی‌گردد. NTFS از دیتاهای تکراری برای فایل سیستم‌های مهم استفاده می‌کند که مشکل از دست رفتن داده را نداشته باشد.
امنیت: از windows object model برای امنیت استفاده می‌کند. هر فایل باز به عنوان یک file object در نظر گرفته می‌شود و ویژگی‌های امنیتی آن فایل مشخص می‌شود. این ویژگی‌های امنیتی به عنوان ویژگی هر فایل در دیسک هستند.

فایل‌ها و دیسک‌های با حجم زیاد: هم حجم بسیار زیاد را ساپورت می‌کند و هم به صورت بهینه مدیریت می‌کند.

چندین data stream: محتوای فایل به عنوان یک جریانی از بایت‌ها در نظر گرفته می‌شود. در NTFS می‌توان چندین جریان داده برای یک فایل تعریف کرد. فایده این مثلاً در استفاده از راه دور در سیستم‌عامل‌های MAC است چون در MAC هر فایل هم داده فایل و اطلاعات فایل را دارد و از این دو به عنوان دو جریان بایتی مختلف برای یک فایل استفاده می‌کند.

ثبت‌وقایع: logهای تمامی تغییرات فایل‌ها را ذخیره می‌کند و اپلیکیشن‌های دیگر می‌توانند این logها و تغییرات را بخوانند.

فشرده‌سازی و رمزنگاری: هر فایل یا دیرکتوری‌ای می‌تواند فشرده و/یا رمزنگاری شود.
لینک‌های hard و symbolic: برای ساپورت POSIX ویندوز لینک‌های hard را ساپورت می‌کند که باعث می‌شود که یک فایل با چند آدرس مختلف در یک volume قابل دسترسی باشد. در symbolic link هم یک فایل یا دیرکتوری با چند آدرس مختلف قابل دسترسی است حتی اگر در volume های مختلف باشند. ویندوز همچنین mount point را ساپورت می‌کند.

۶)

یک دیسک جدید و خالی فقط دارای سخت‌افزار است، قبل از این که بتواند برای ذخیره داده استفاده شود باید به sectorهای مختلفی تقسیم شود. این فرایند فرمت سطح پایین یا فرمت فیزیکی نامیده می‌شود. فرمت سطح پایین دیسک را با ساختمان داده مخصوصی برای هر سکتور پر می‌کند. این ساختمان داده شامل سرآیند، بخش داده و trailer است. سرآیند و انتها حاوی داده‌هایی برای دیسک کنترلر هستند که برای مثال شماره sector را مشخص کند یا بیت‌های تصحیح خطا در آن قرار گیرند.

بیشتر دیسک‌ها در کارخانه فرمت سطح پایین می‌شوند. این کار تولیدکننده را قادر به تست دیسک و شروع تبدیل آدرس از شماره بلاک منطقی به سکتور در دیسک می‌کند.

سیستم‌عامل قبل از استفاده از دیسکدو کار مقابل را انجام می‌دهد: پارتیشن بندی و فرمت منطقی

۷)

چون head دیسک حرکت می‌کند در معرض خطا است. اگر این خطا کامل و زیاد باشد، در بعضی مواقع دیسک نیاز به جایگزینی دارد و داده‌های آن روی یک دیسک جدید بازیابی می‌شوند. در اکثر مواقع فقط یک یا چند sector دچار مشکل می‌شوند. خیلی از دیسک‌ها حتی از کارخانه هم با بلاک بد تولید می‌شوند. این بلاک‌های بد بسته به دیسک و نوع کنترلر در حالت‌های متفاوتی مدیریت می‌شوند.

روی دیسک‌های ساده بلاک‌های بد به صورت دستی مدیریت می‌شوند. یک استراتژی این است که کل دیسک زمانی که در حال فرمت است برای یافتن بلاک‌های بد اسکن شود. هر بلاک بدی که پیدا شد با فلگ unusable علامت‌گذاری شود که فایل سیستم‌ها دیگر از آن استفاده نکنند. اگر این بد بلاک‌ها در حین عملیات عادی به وجود بیایند باید یک برنامه این بد بلاک‌ها را جستجو و لاک کند. داده‌های روی بد بلاک عموما از بین می‌روند.

دیسک‌های پیچیده‌تر هوشمندتر هستند. آن‌ها لیست بلاک‌های بد را نگه می‌دارند. این لیست در هنگام فرمت سطح پایین در کارخانه مقداردهی اولیه شده و همواره به‌روزرسانی می‌شود.

فرمت سطح پایین تعدادی سکتور یدکی کنار می‌گذارد که برای سیستم‌عامل قابل مشاهده نیست. کنترلر می‌تواند هر سکتور بد را به صورت منطقی با یکی از همین سکتورهای یدکی جایگزین کند. به این کار sector sparing می‌گویند.

به عنوان یک جایگزین برای sector sparing، برخی از کنترلرها می‌توانند بلاک بد را به وسیله sector slipping جایگزین کنند. روش هم به گونه‌ای است که انگار شیفت داده می‌شوند.

جایگزینی داده در بد بلاک هم عموما اتوماتیک نیست چون داده‌ها عموما از دست می‌روند. (سافت ارور برای زمانی که کاملاً از دست نرفته و یک کپی داریم و هارد ارور برای دیتای کاملاً از دست رفته)

۸)

برای شروع اجرای یک کامپیوتر باید یک برنامه‌ای برای اجرای اولیه باشد. این برنامه ساده بوت‌سترپ تمامی بخش‌های سیستم مانند رجیسترهای پردازنده و محتوای حافظه اصلی را مقدار دهی اولیه می‌کند و سپس سیستم‌عامل را اجرا می‌کند. برای این کار کرنل را پیدا و لود کرده و به نقطه شروع آن می‌رود.

برای این که برای تغییر بوت‌سترپ به مشکل نخوریم یک قسمت از آن روی ROM قرار می‌گیرد که بخش دیگری از همین برنامه را از روی DISK لود می‌کند.

برنامه بوت‌سترپ در **بلاک بوت** در یک مکان ثابت از دیسک دخیره می‌شود. یک دیسک که پارتیشن بوت داشته باشد بوت دیسک یا سیستم دیسک نامیده می‌شود.

بوت پارتیشن تمامی سیستم‌عامل، بوت و درایورهای دیوایس‌ها را دارد. ویندوز کد بوت خود را در اولین سکتور هارد دیسک می‌گذارد که **MBR** نامیده می‌شود. عملیات بوت با اجرای کد داخل ROM شروع می‌شود و باعث می‌شود کد بوت را از MBT بخوانیم. MBR علاوه بر داشتن کد بوت شامل جدولی است که پارتیشن‌های هارد دیسک و یک فلگ که نشان می‌دهد که سیستم از کدام پارتیشن بوت شود است. زمانی که بوت پارتیشن مشخص شد، سیستم اولین سکتور را از بوت پارتیشن می‌خواند که **بوت سکتور** نامیده می‌شود و ادامه فرایند بوت را که شامل لود ساب سیستم‌ها است را انجام می‌دهد.

۹)

اندروید از قابلیت‌های مدیریت فایل لینوکس استفاده می‌کند. دیرکتوری فایل سیستم در اندروید مشابه لینوکس است البته با کمی تغییر و شخصی سازی اندرویدی.

دیرکتوری سیستم (آدرس system/) دارای بخش‌های اصلی سیستم‌عامل مانند فایل‌های باینری، کتابخانه‌ها و ... است. این دیرکتوری همچنین شامل برخی از نرم‌افزارهای پیش‌فرض اندرویدی است. این دیرکتوری read only است و کاربران فقط می‌توانند آن را بخوانند.

دیرکتوری داده (آدرس data/) برای ذخیره کردن داده‌های مختص اپلیکیشن و شخصی اپلیکیشن استفاده می‌شود. زمانی که کاربر گوشی را ریست کارخانه می‌کند این پارتیشن دوباره ایجاد می‌شود و قبلی پاک می‌شود.

زمانی که برنامه‌ای نصب می‌شود ابتدا فایل .apk آن در بخش data/app/ قرار داده شده و یک ساب‌دیرکتوری در دیرکتوری data/data/ ساخته می‌شود که فقط خود برنامه به آن دسترسی دارد.

دیرکتوری کش در آدرس cache/ قرار دارد و برای ذخیره‌سازی موقت توسط سیستم‌عامل استفاده می‌شود. سیستم‌عامل برای دسترسی سریع‌تر به داده‌ها از این دیرکتوری استفاده می‌کند. پاک کردن این دیرکتوری تأثیری روی داده‌های کاربران ندارد.

دیرکتوری mnt/sdcard/ یک پارتیشن SD کارت است که در تعامل با دیوایس ما می‌باشد. این فضا برای نوشتن/خواندن داده کاربر و فایل‌های چندرسانه‌ای کاربر استفاده می‌شود.

۱۰)

CD برای ذخیره صوت ساخته شده بود که می‌تواند بیش از ۶۰ دقیقه صوت را در خود نگه دارد. نوشتن تکنیکال به این صورت است که برای بیت‌های خاصی یک نقطه سوزانده می‌شود و این کار باعث می‌شود نور تابیده شده با شدت کم‌تری بازتاب شود. اگر آن نقطه سوزانده یا دست‌کاری نشود نشان‌دهنده یک بیت مخالف است. خواندن از روی آن نیز به صورت تابش و بازتاب است که اگر شدت بازتاب کم باشد بیت ۱ و اگر زیاد باشد بیت مخالف آن را نشان می‌دهد. برای افزایش ظرفیت هم می‌توان با فاصله گرفتن از مبدأ دوباره تراکم بیت‌ها را بیشتر کرد.

DVD اما برای تبدیل ویدیو به حالت دیجیتال و ظرفیت بیشتر تولید شد و حتی به جای CD نیز استفاده می‌گردد. ظرفیت بیشتر DVD به سه دلیل است:

- بیت‌ها نسبت به CD به هم نزدیک‌تر هستند و تراکم بیش‌تری دارند.
- DVD می‌تواند یک لایه بیشتر برای ذخیره‌سازی داشته باشد که ظرفیت آن را دوبرابر می‌کند و می‌توان با تغییر فکوس لیزر بین لایه‌های مختلف switch کرد.
- DVD می‌تواند در هر دو سمت آن نوشته و خوانده شود که ظرفیت آن را دوبرابر می‌کند.

Blu-Ray DVDها ظرفیتی حتی بیش‌تر از DVD دارد که به خاطر تراکم بیتی بیشتر است. برای این که تراکم بیتی بیش‌تر داشته باشیم باید یک لیزر با طول موج کم‌تری داشته باشیم(در محدوده بنفش). Pitهای داده (نقاط ۰ و ۱) نیز کوچکتر هستند.

۱۱)

در بین سال‌های ۱۹۵۰ تا ۱۹۷۰ میلادی کاربرد داشت و به صورتی که یک ورقه فلزی که لایه خارجی آن با مواد فرومغناطیسی پوشانده شده است می‌چرخید و برای رسیدن به یک نقطه باید rotational delay را صبر می‌کردیم. برای خواندن و نوشتن چندین نوک وجود دارد که هر شیار به تعدادی سکتور تقسیم می‌شود.

چون این نوع رسانه ذخیره سازی از دیسک سریع‌تر است (seek time ندارد) ولی ظرفیت کم‌تری دارد می‌تواند به عنوان حافظه نهان استفاده شود چون به یک حافظه سریع و کم حجم نیاز داریم.

سرعت زیاد حافظه طبلی به دلیل نیاز نداشتن جابه‌جایی head و ثابت بودن آن و در نتیجه ۰ شدن seek time است.