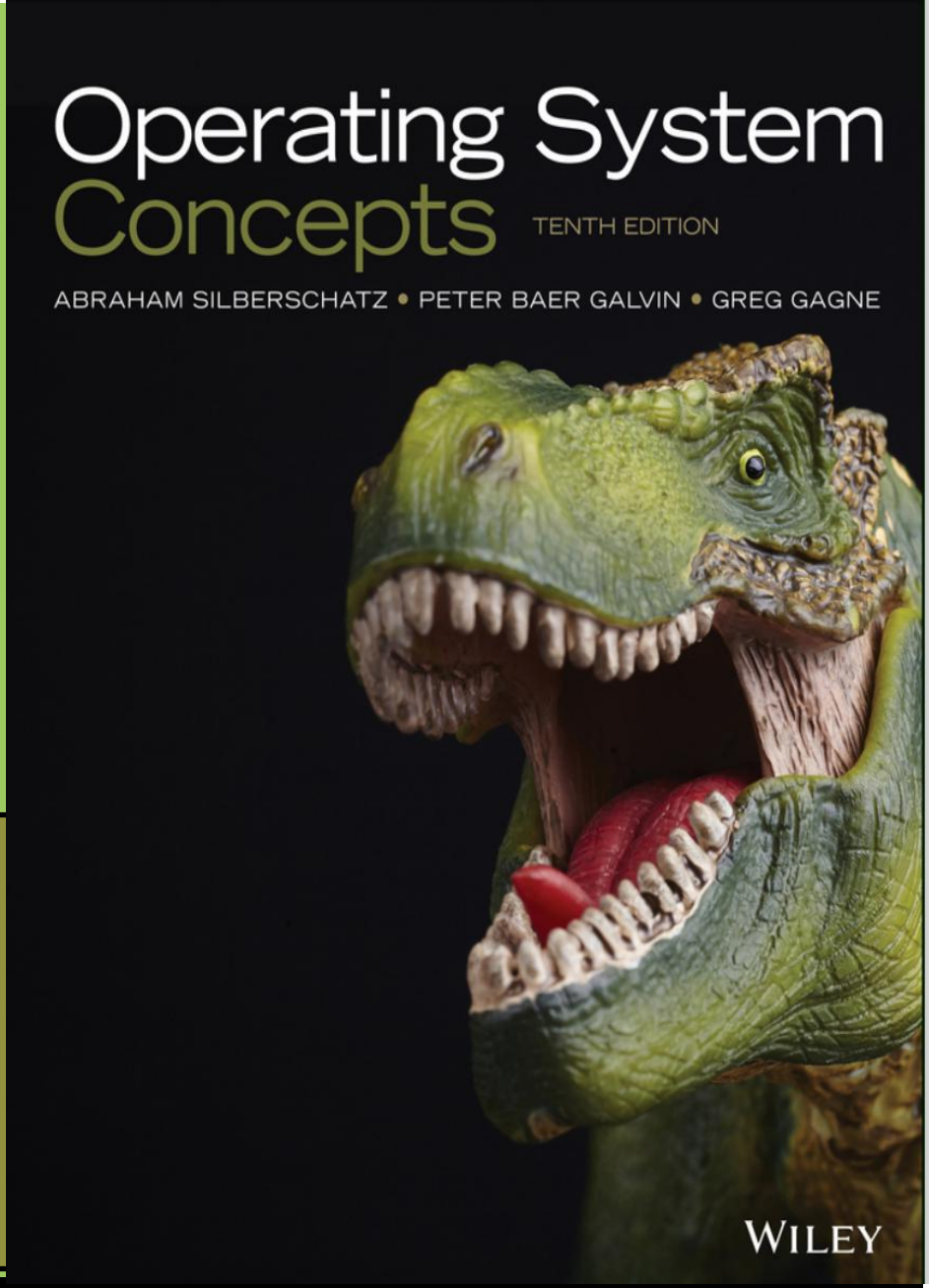
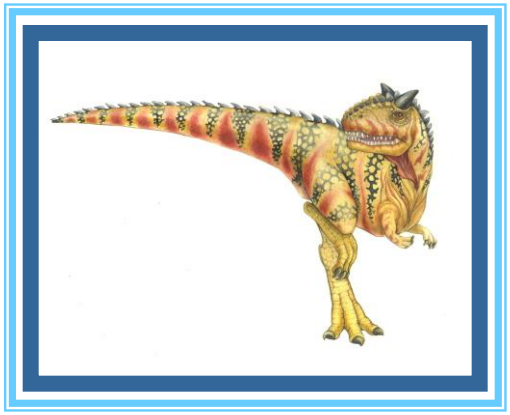
سیستم‌عامل

Dr. A. Taghinezhad



Website: ataghinezhad.github.io, Email:a0taghinezhad@gmail.com

فصل ۱۰: سیستم‌های ذخیره‌سازی حجیم



Objectives اهداف

- ساختار فیزیکی دستگاه‌های ذخیره‌سازی ثانویه و تأثیر ساختار بر کاربردها:
دستگاه‌های ذخیره‌سازی ثانویه مثل هارد دیسک‌ها، SSDها و نوارهای مغناطیسی، از بخش‌های مختلفی مثل دیسک‌ها، هدها، مدارهای کنترل و حافظه‌های فلش تشکیل شده‌اند. ساختار فیزیکی هر دستگاه تعیین می‌کند که سرعت خواندن/نوشتن، دوام، ظرفیت و نحوه استفاده‌اش چگونه باشد. مثلا هارد دیسک‌ها به‌خاطر قطعات مکانیکی‌شان کندتر ازSSD ها هستند ولی ظرفیت بالاتری دارند.

- ویژگی‌های عملکردی دستگاه‌های ذخیره‌سازی حجیم:
سرعت انتقال داده، زمان دسترسی، نرخ خطا، و مصرف انرژی از مهم‌ترین ویژگی‌ها هستندSSD .ها سرعت و دوام بالاتری دارند، ولی گران‌ترند؛ هارد دیسک‌ها ارزان‌ترند ولی کندتر و آسیب‌پذیرتر.

- ارزیابی الگوریتم‌های زمان‌بندی ورودی/خروجی:(I/O scheduling)
الگوریتم‌هایی مثلFCFS ، SSTF، SCANو C-SCAN برای بهینه‌سازی زمان دسترسی و کارایی استفاده می‌شوند. این الگوریتم‌ها ترتیب درخواست‌ها را طوری تنظیم می‌کنند که هد دستگاه حداقل جابجا شود و عملکرد افزایش یابد.

پیش زمینه

بیشتر حافظه‌های جانبی توی کامپیوترهای امروزی از هارد دیسک (HDD) و حافظه‌های غیرفرّار (NVM) هستن.

HDDیه دیسک چرخون با پوشش مغناطیسی داره که زیر هدهای خوندن/نوشتن حرکت می‌کنه.

- سرعت چرخش دیسک حدود ۶۰ تا ۲۵۰ دور در ثانیه‌ست.
- نرخ انتقال یعنی سرعت جابه‌جایی داده بین دیسک و کامپیوتر.
- زمان دسترسی تصادفی شامل دو بخشه: زمان حرکت بازوی دیسک (seek time) و زمان چرخیدن تا رسیدن سکتور مورد نظر.(rotational latency)
- اگه هد با سطح دیسک برخورد کنه، بهش می‌گن "head crash" که خیلی بده و ممکنه داده‌ها از بین برن.

بعضی دیسک‌ها قابل جداشدنن.

Moving-head Disk Mechanism

Platterدیسک گردی که داده‌ها روش ذخیره می‌شن. چندتا از اینا رو هم می‌ذارن.

Spindleمیله‌ای که دیسک‌ها بهش وصلن و دور خودش می‌چرخه.

Track (t):یه دایره‌ی نازک روی دیسک که داده‌ها توی اون ذخیره می‌شن.

Sector (s):بخش کوچکی از یه ترک، مثل یه تکه پازل از داده.

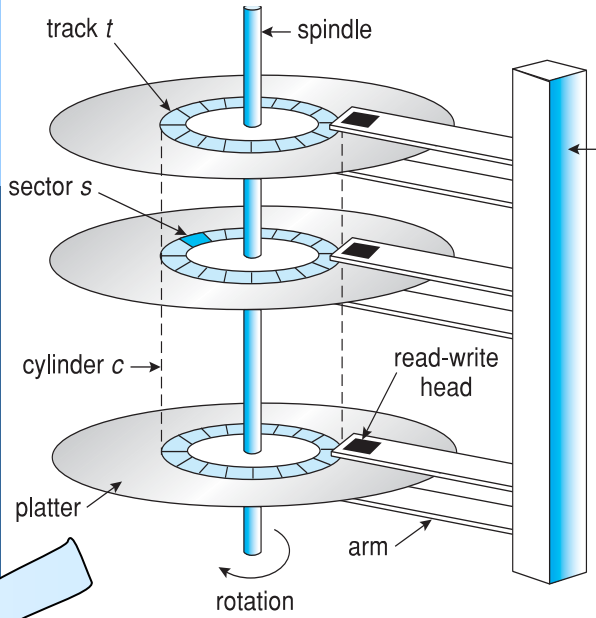
Cylinder (c):مجموعه ترک‌های هم‌ردیف توی دیسک‌های مختلف.

Read-write head:همون چیزی که داده‌ها رو می‌خونه یا می‌نویسه.

Arm:بازویی که هد رو جابه‌جا می‌کنه تا به ترک‌های مختلف برسه.

Arm assembly:قسمت مکانیکی که این بازوها رو کنترل می‌کنه.

Rotation:جهت چرخیدن دیسک‌ها.



زمان‌بندی دیسک سخت(HDD Scheduling)

سیستم‌عامل باید از سخت‌افزار به‌صورت مؤثر استفاده کند. برای دیسک‌ها یعنی:

- کاهش زمان جستجو(Seek Time)
  هر چه فاصله‌ی جستجو کمتر باشد، زمان دسترسی سریع‌تر است.
- افزایش پهنای باند دیسک
  یعنی حجم اطلاعات منتقل‌شده نسبت به زمان کلی انتقال.

مدیریت درخواست‌های دیسک

- درخواست‌های I/O ممکن است از طرف:سیستم‌عامل-فرایندهای سیستمی-برنامه‌های کاربر باشند
- هر درخواست شامل: نوع ورودی/خروجی، آدرس دیسک، آدرس حافظه، تعداد سکتورها است.
- اگر دیسک بیکار باشد، بلافاصله اجرا می‌شود.
  در غیر این‌صورت در صف منتظر می‌ماند.
- الگوریتم‌های بهینه‌سازی فقط وقتی معنا دارند که صف وجود داشته باشد.

وضعیت امروزی: قبلاً زمان‌بندی هد توسط سیستم‌عامل انجام می‌شد.
الان در خود سخت‌افزار و کنترلرهای ذخیره‌سازی انجام می‌شود.

- فقط آدرس منطقی بلوک (LBA) داده می‌شود، کنترلر بقیه را انجام می‌دهد.

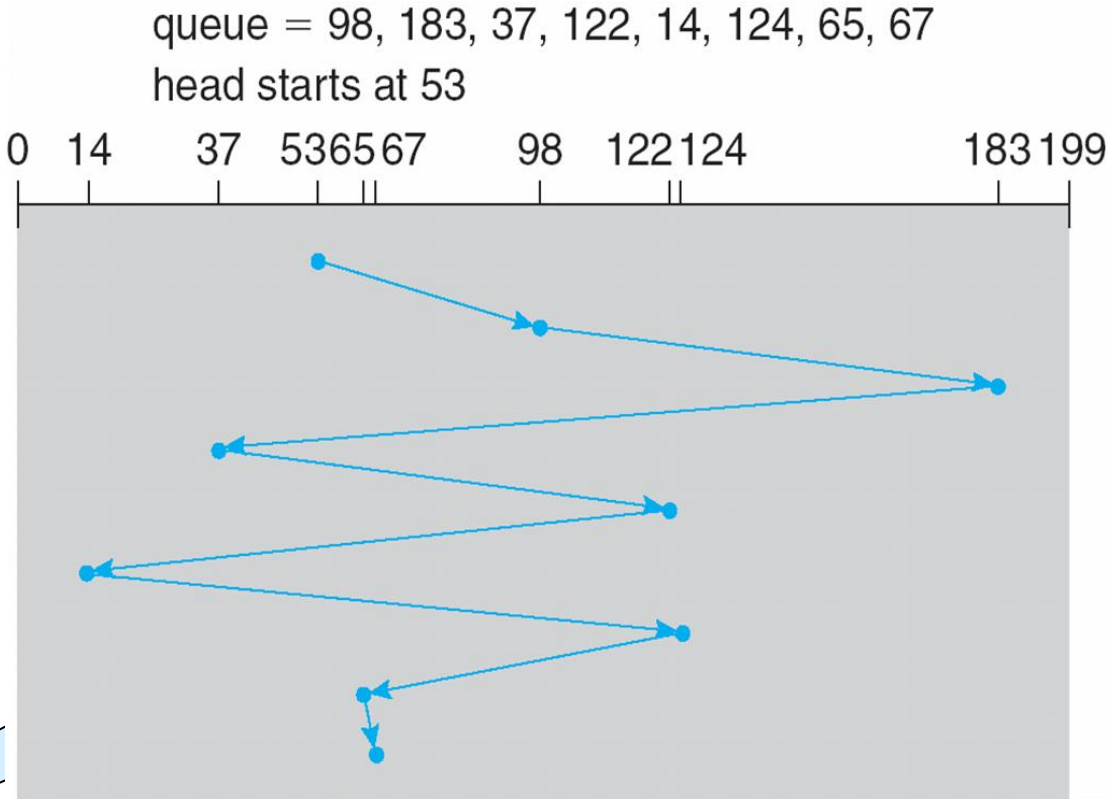
زمان‌بندی دیسک

الگوریتم‌ها

- کنترلر دیسک یک بافر کوچک دارد و صفی از درخواست‌ها را مدیریت می‌کند.
- الگوریتم‌های مختلفی برای زمان‌بندی این درخواست‌ها وجود دارد.
- این تحلیل برای دیسک‌هایی با یک یا چند پلاتر معتبر است.
- مثال: صف درخواست‌ها بین شماره‌های ۰ تا ۱۹۹
  صف: ۹۸, ۱۸۳, ۳۷, ۱۲۲, ۱۴, ۱۲۴, ۶۵, ۶۷
  موقعیت فعلی هد53 :

FCFS

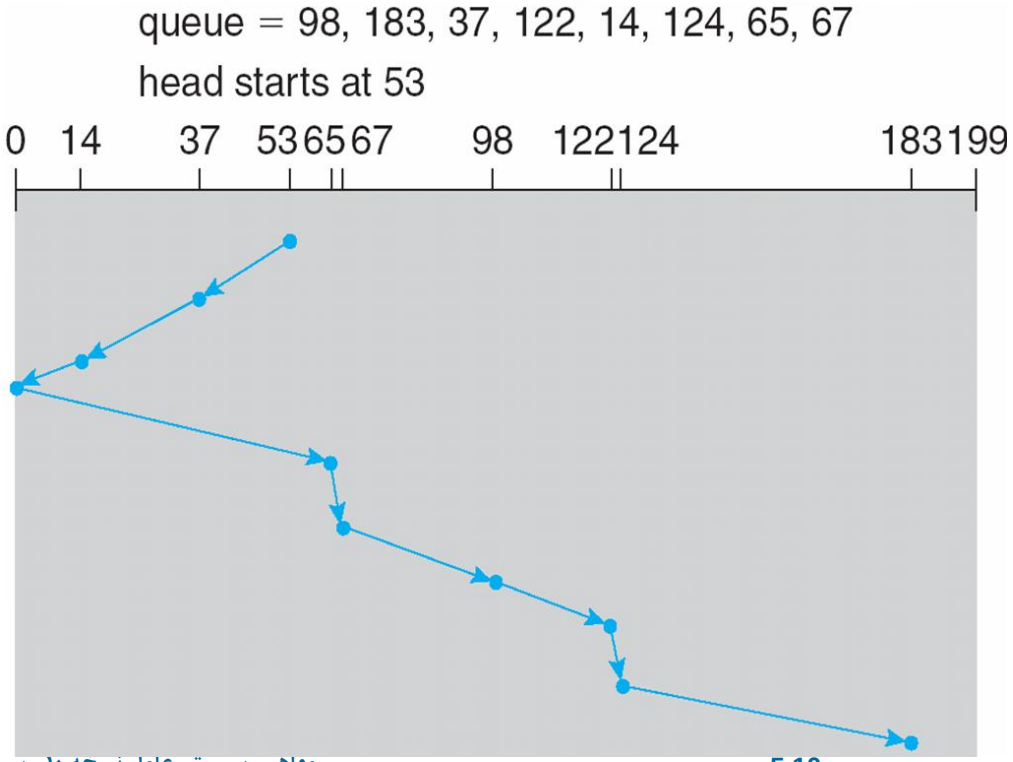
تصویر نشان‌دهنده‌ی حرکت هد دیسک با الگوریتم **FCFS** است.
در این روش، درخواست‌ها به ترتیبی که وارد صف شده‌اند اجرا می‌شوند، بدون هیچ بهینه‌سازی.



- مجموع حرکت هد: ۶۴۰ سیلندر
- یعنی هد برای سرویس دادن به همه‌ی درخواست‌ها، مجموعاً ۶۴۰ واحد حرکت کرده.
- حرکت زیاد هد = سرعت پایین‌تر و تأخیر بیشتر.

SCAN

- در **الگوریتم SCAN** ، هد دیسک مثل آسانسور عمل می‌کند:
  - از یک سمت شروع می‌کند (اینجا از موقعیت 53به سمت چپ)تمام درخواست‌های سر راهش را انجام می‌دهد
  - سپس جهت را عوض می‌کند و در مسیر برگشت، بقیه‌ی درخواست‌ها را سرویس می‌دهد
- **مجموع حرکت هد** 208 سیلندر
- خیلی بهینه‌تر از FCFS که ۶۴۰ بود



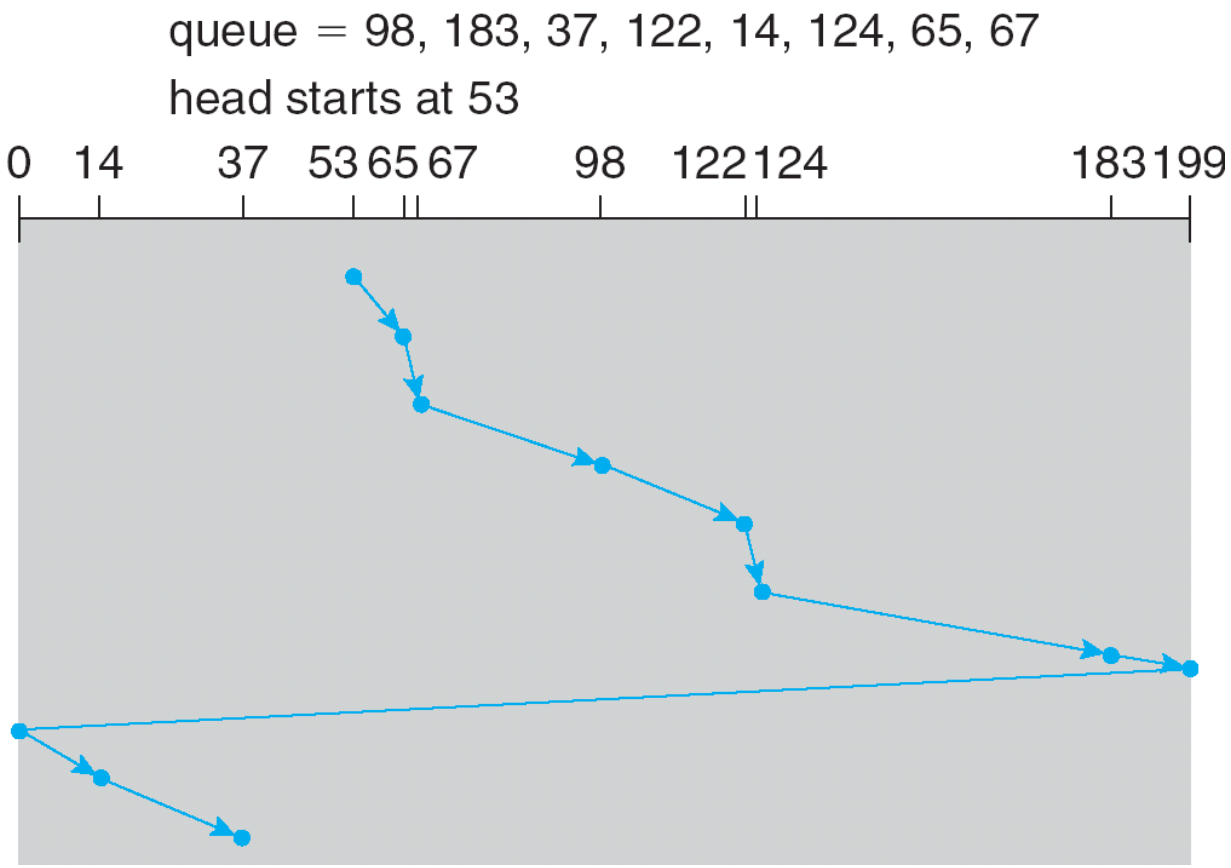
C-SCAN

- در **C-SCAN**، هد دیسک فقط در یک جهت درخواست‌ها را بررسی می‌کند (مثلاً از چپ به راست).
- وقتی به انتهای دیسک رسید، بدون انجام هیچ کاری، سریع به ابتدای دیسک برمی‌گردد و دوباره شروع می‌کند.

ویژگی‌ها:

- زمان انتظار **یکنواخت‌تر** نسبت به SCAN فراهم می‌کند.
- دیسک مثل یک **لیست حلقه‌ای** در نظر گرفته می‌شود.
- در مسیر برگشت، هد **هیچ درخواستی را انجام نمی‌دهد.**

C-SCAN (Cont.)



تعداد کل سیلندرها = ۲۰۰

Selecting a Disk-Scheduling Algorithm

📌 انتخاب الگوریتم زمان‌بندی دیسک(Disk Scheduling)

- SSTF(نزدیک‌ترین درخواست): ساده و محبوب است، اما ممکن است منجر به گرسنگی (starvation) شود.
- SCANو C-SCAN برای سیستم‌هایی با بار سنگین روی دیسک بهترند.
  - گرسنگی کمتر، اما باز هم ممکن است رخ دهد.

مدیریت دستگاه ذخیره‌سازی (ادامه)

- فرمت سطح پایین:(Low-Level Formatting)
  - تقسیم دیسک به سکتورها شامل داده، اطلاعات سربرگ، وECC
  - معمولاً ۵۱۲ بایت، ولی قابل تغییر است.
- فرمت منطقی یا ساخت فایل‌سیستم:
  - دیسک به پارتیشن‌هایی با سیلندرهای مجزا تقسیم می‌شود.
  - فایل‌سیستم‌ها برای بهره‌وری، بلاک‌ها را به کلاستر گروه‌بندی می‌کنند.
  - I/Oدیسک با بلاک، ولی فایل I/O با کلاستر انجام می‌شود.

Storage Device Management (cont.)

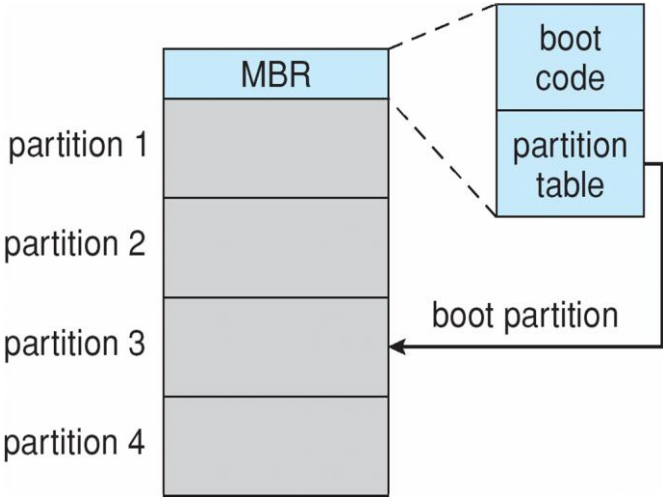
- پارتیشن ریشه (Root) شامل سیستم‌عامل است.
  - دیگر پارتیشن‌ها می‌توانند سیستم‌عامل‌های دیگر یا فایل‌سیستم‌های متفاوت یا حتی فضای خام (raw) باشند.
- هنگام mountکردن:
  - صحت اطلاعات فایل‌سیستم بررسی می‌شود(metadata)
  - اگر درست بود، mountمی‌شود، اگر نه تلاش دوباره یا اصلاح انجام می‌شود.
- Boot Block:
  - می‌تواند به سیستم‌عامل یا boot loader اشاره کند.
  - یا برنامه‌ای برای راه‌اندازی چند سیستم‌عامل باشد.

راه‌اندازی سیستم از دیسک(Booting)

- دسترسی خام به دیسک:
  - برخی برنامه‌ها مانند پایگاه‌های داده (Databases) ترجیح می‌دهند مستقیماً به دیسک دسترسی داشته باشند.
  - در این حالت، سیستم‌عامل کنار گذاشته می‌شود و برنامه خودش مدیریت بلاک‌ها را انجام می‌دهد.
- Boot Block بلوک راه‌انداز
  - برای راه‌اندازی سیستم (Boot) استفاده می‌شود.
  - کد بوت‌استرپ (bootstrap code) در ROM یا firmware ذخیره می‌شود.
  - کد لودر بوت‌استرپ (Bootstrap Loader) در boot block واقع در boot partition ذخیره می‌شود.
    - وظیفه آن: بارگذاری کرنل از دیسک به حافظه و شروع اجرای سیستم‌عامل.

Device Storage Management (Cont.)

- **MBR (Master Boot Record):** اولین سکتور دیسک (معمولاً ۵۱۲ بایت)
  - شامل **Boot Code:** کدی که اجرا می‌شود تا سیستم‌عامل را راه‌اندازی کند.
  - **Partition Table:** اطلاعات مربوط به ۴ پارتیشن اصلی دیسک را نگه می‌دارد.
  - **MBR** مشخص می‌کند کدام پارتیشن بوت است.



تصویر بالا ساختار ابتدایی دیسک را نشان می‌دهد که شامل MBR (Master Boot Record) و چهار پارتیشن است.

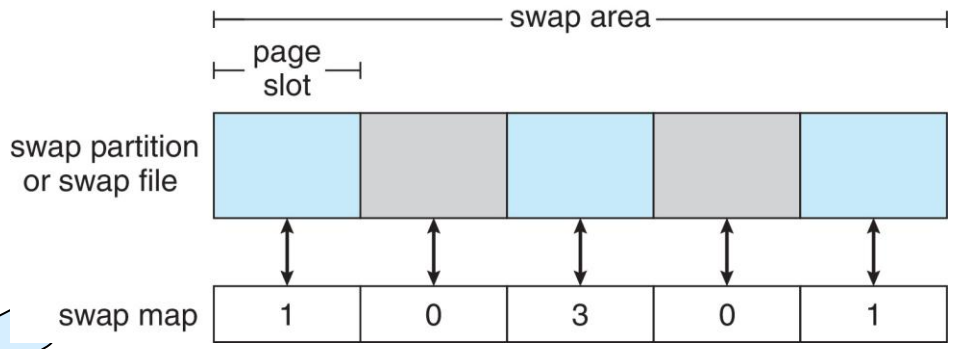
مدیریت بلوک‌های خراب:

- روش‌هایی مانند **sector sparing** برای مقابله با سکتورهای خراب به‌کار می‌رود:
  - سکتور خراب از دور خارج شده و سکتور جایگزین اختصاص داده می‌شود.
  - معمولاً توسط سخت‌افزار دیسک (firmware) انجام می‌شود.

Swap-Space Management

مدیریت فضای Swap : وقتی حافظه اصلی (DRAM) برای اجرای تمام فرایندها کافی نیست، سیستم‌عامل از **swap space** برای ذخیره موقت داده‌ها (کل فرایند یا صفحات مجزا) در دیسک استفاده می‌کند.

- **Swap Partition / Swap File:** محل فیزیکی ذخیره داده‌های swap.
  - می‌تواند یک **پارتیشن جداگانه** باشد (سریع‌تر)، یا یک **فایل معمولی** در سیستم فایل (راحت‌تر برای افزودن).
  - به قسمت‌های مساوی به نام **page slot** تقسیم می‌شود.
- **Swap Map:** یک آرایه‌ای از اعداد است که مشخص می‌کند هر **page slot** چند بار استفاده شده یا آزاد است.
  - عدد **0** یعنی آن slot خالی است.
  - غیر صفر در حال استفاده بودن است



End of Chapter