یادداشت‌های درس سیستم‌های عامل - بخش سیزدهم

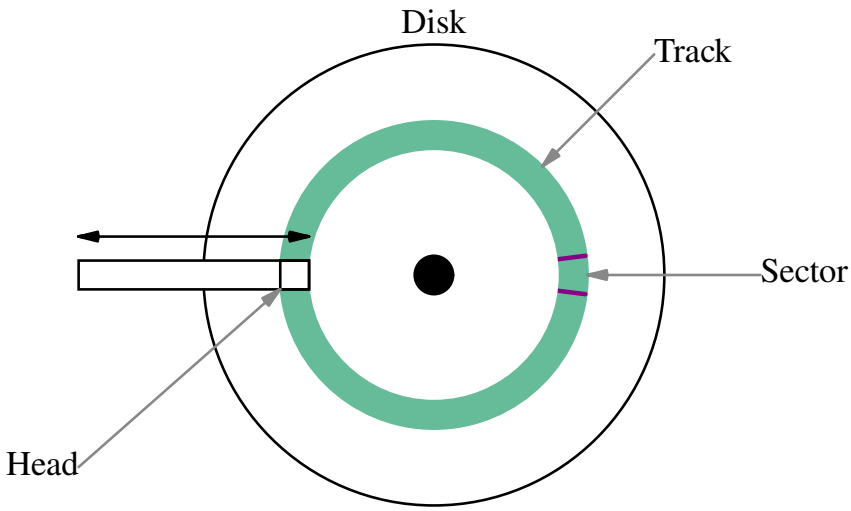
در این قسمت از درس مطالبی در مورد دیسک‌ها و زمانبندی آنها را مطالعه می‌نماییم.

در یکی از جلسه‌های شروع درس در مورد دیسک و فایل سیستم مطالبی را مطرح کردم.

- امروزه دو تکنولوژی به صورت گسترده برای ذخیره‌سازی اطلاعات استفاده می‌شوند: دیسک‌های مغناطیسی (Magnetic Disk) و دیسک‌های حالت ایسا (Solid State Disks یا به صورت مخفف SSD).
- با این دو نوع دیسک در ادامه آشنا می‌شویم.

در دیسک‌های مغناطیسی تعداد دیسک (سطح دایره‌ای) قرار دارند که موازی با هم حول یک محور قرار گرفته‌اند.

- سطح دیسک به تعدادی ناحیه‌ی دایره‌ای موازی به نام شیار (Track) تقسیم می‌شود.
- هر شیار به تکه‌های کوچکی به نام سکتور (Sector) تقسیم می‌شود که قسمتی از اطلاعات دیسک (مثل یک کیلو بایت) را نگه می‌دارد.
- روی هر دیسک دسته‌ای (Arm) متحرک وجود دارد. روی نوک این دسته سر خواندن و نوشتن (Head) قرار دارد که می‌تواند اطلاعات یک Sector را بخواند. طول دسته‌ی دیسک تغییر می‌کند تا روی Track مشخصی قرار گیرد و دیسک نیز می‌چرخد تا یک Sector مشخص زیر آن قرار گیرد.
- بنابراین، برای دسترسی به اطلاعات دیسک، در هر درخواست دسترسی به دیسک شماره‌ی Track و شماره‌ی Sector اطلاعات مشخص می‌شود.
- شکل زیر شمای یک دیسک و سر خواندن آن را از بالا نشان می‌دهد.



بنابراین برای دسترسی به اطلاعات روی دیسک دو گام انجام می‌شود.

الف) طول دسته‌ی سر خواندن و نوشتن کوتاه یا بلند می‌شود تا Head روی Track مورد نظر قرار گیرد.

ب) دیسک می‌چرخد تا دیسک روی Sector مورد نظر قرار بگیرد.

- به زمان مورد نیاز برای اینکه Head روی Sector مورد نظر قرار گیرد زمان Positioning یا دسترسی تصادفی (Random-access time) گفته می‌شود.
- به زمانی که لازم است تا Head روی Track مورد نظر قرار گیرد Seek time و به زمانی که لازم است تا دیسک بچرخد تا Head رو Sector مورد نظر قرار گیرد تأخیر چرخشی (Rotational Latency) گفته می‌شود.
- بنابراین زمان دسترسی تصادفی دیسک برابر مجموع Seek time و Rotational Latency هست.

دیسک‌های حالت ایستا

- دیسک‌های حالت ایستا یا SSD با استفاده از تکنولوژی مشابه دیسک‌های Flash امکان ذخیره‌سازی اطلاعات را بدون قسمت متحرک فراهم می‌کنند.
- چون قسمت متحرکی ندارند، معمولا تأخیر کمتری نسبت به دیسک‌های مغناطیسی دارند و سریع‌تر هستند.
- چون قسمت متحرکی ندارند، معمولا قابلیت اطمینان بیشتری نسبت به دیسک‌های مغناطیسی دارند.
- چون قسمت متحرکی ندارند، معمولا انرژی کمتری نسبت به دیسک‌های مغناطیسی مصرف می‌کنند.
- معمولا دیسک‌های SSD به علت محدودیت تعداد دفعات نوشتن روی هر بلوک، عمر کمتری نسبت به دیسک‌های مغناطیسی دارند.

در یک سیستم عامل ممکن است تعداد زیادی درخواست از طرف پردازه‌های مختلف برای دسترسی به بلوک‌های دیسک به سیستم عامل داده شود.

- سیستم عامل برای پاسخ به این درخواست‌ها آنها را به دیسک می‌فرستد تا عملیات ورودی یا خروجی مناسب انجام شود.
- به تعیین ترتیب فرستادن درخواست‌ها به دیسک، زمانبندی دیسک (Disk Scheduling) می‌گویند. در ادامه زمانبندی دیسک‌های مغناطیسی را بررسی می‌کنیم.

همانطور که اشاره شد، در دیسک‌های مغناطیسی هر درخواست به Track و Sector مشخصی از یک دیسک دسترسی دارد.

- در هر لحظه ممکن است درخواست‌های متفاوتی برای دسترسی به قسمت‌ها مختلف دیسک به سیستم عامل داده شده باشد.
- سرعت چرخش دیسک معمولا چند هزار دور در دقیقه است و توسط شرکت سازنده‌ی آن اعلام می‌شود و سیستم عامل نمی‌تواند تأثیر زیادی در بهبود Rotational Latency داشته باشد.
- اما سیستم عامل می‌تواند با تعیین ترتیب مناسب برای دسترسی درخواست‌ها میزان حرکت دسته‌ی Head و در نتیجه Seek-time را کاهش دهد و در نتیجه سرعت پاسخ به درخواست‌ها را افزایش دهد.

الگوریتم‌های زمانبندی دیسک

در ادامه فرض می‌کنید تعدادی درخواست دسترسی به قسمت‌های مختلف دیسک به سیستم عامل داده شده است و سیستم عامل باید ترتیب فرستادن آنها را به دیسک مشخص نماید.

- این کار توسط زمانبند دیسک انجام می‌شود.
- چون فقط شماره‌ی Track اهمیت دارد، در مثال‌ها فقط شماره‌ی Track درخواست‌ها بیان می‌شود.
- فرض کنید به ترتیب به درخواست‌های زیر دسترسی انجام می‌شد.

Request	1	2	3	4	5	6	7	8	9	10
Track #	45	70	56	10	22	90	33	85	80	75

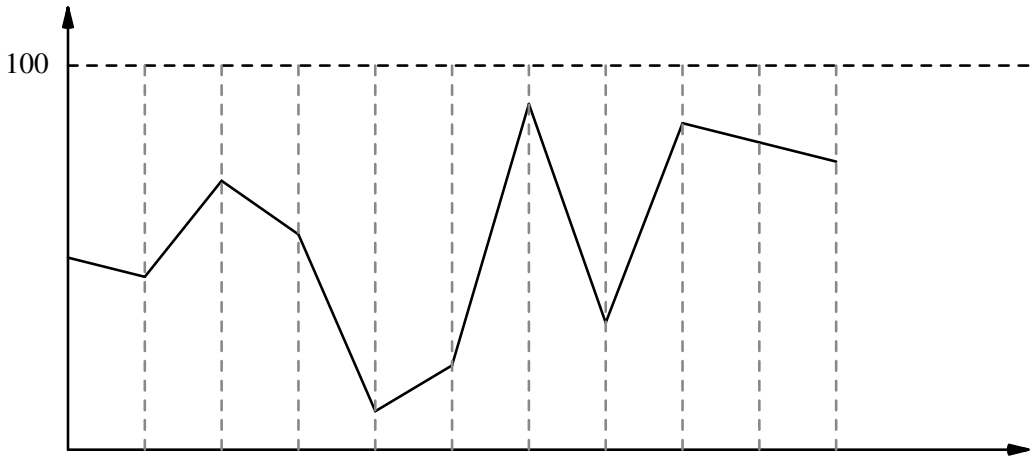
الگوریتم FCFS

در الگوریتم FCFS (First-Come First-Served) درخواست‌ها با توجه به ترتیب ورود آنها به صف زمانبندی دیسک انجام می‌شوند.

- فرض کنید در ابتدا Head روی Track شماره‌ی پنجاه باشد و به سمت Track شماره‌ی صد در حرکت باشد؛ درخواست‌های جدول زیر را در نظر بگیرید.

Request	1	2	3	4	5	6	7	8	9	10
Track #	45	70	56	10	22	90	33	85	80	75

- در شکل زیر، ترتیب فرستاده شدن درخواست‌ها به دیسک (مکان Head) نشان داده شده است.



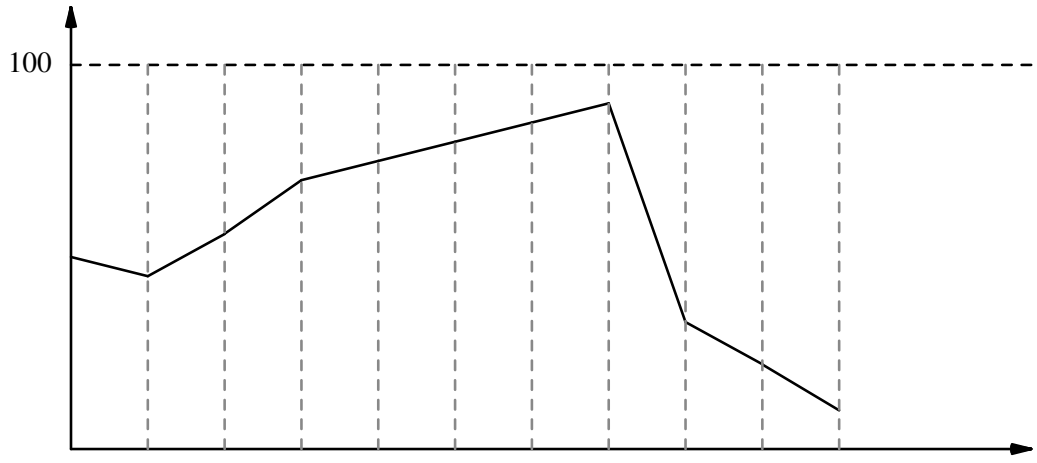
الگوریتم SSTF

در الگوریتم SSTF (Shortest Seek-Time First) پس از هر درخواست به سراغ نزدیک‌ترین درخواست موجود می‌رویم (درخواست که شماره‌ی Track آن اختلاف کمتری نسبت به شماره‌ی Track سر خواندن و نوشتن داشته باشد).

- فرض کنید در ابتدا Head روی Track شماره‌ی پنجاه باشد و به سمت Track شماره‌ی صد در حرکت باشد؛ درخواست‌های جدول زیر را در نظر بگیرید.

Request	1	2	3	4	5	6	7	8	9	10
Track #	45	70	56	10	22	90	33	85	80	75

- در شکل زیر، ترتیب فرستاده شدن درخواست‌ها به دیسک (مکان Head) نشان داده شده است.



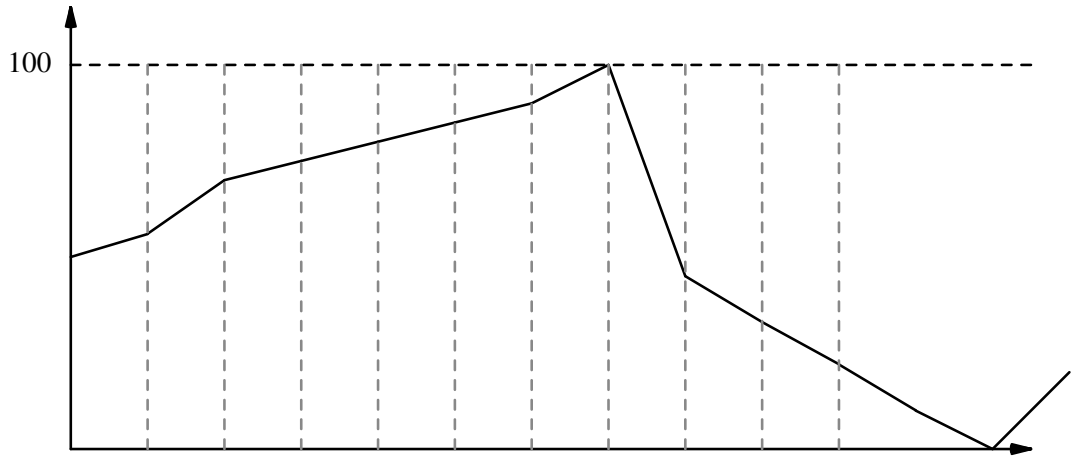
الگوریتم بالابر

در الگوریتم بالابر (Elevator) یا SCAN سر نوشتن به یک سمت حرکت می‌کند و به هر درخواستی که در این مسیر وجود دارد پاسخ می‌دهد. سپس بر می‌گردد و در مسیر برگشت به درخواست‌ها پاسخ می‌دهد و همین کار ادامه پیدا می‌کند.

- فرض کنید در ابتدا Head روی Track شماره‌ی پنجاه باشد و به سمت Track شماره‌ی صد در حرکت باشد؛ درخواست‌های جدول زیر را در نظر بگیرید.

Request	1	2	3	4	5	6	7	8	9	10
Track #	45	70	56	10	22	90	33	85	80	75

- در شکل زیر، ترتیب فرستاده شدن درخواست‌ها به دیسک (مکان Head) نشان داده شده است.

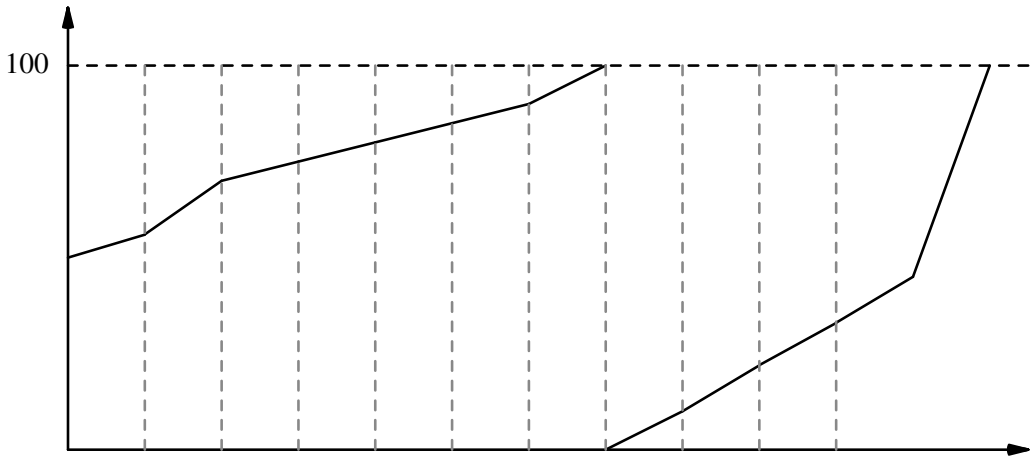


الگوریتم C-SCAN (Circular SCAN) مشابه الگوریتم SCAN هست با این تفاوت که وقتی سر خواندن به آخرین Track رسید حرکتش را دوباره از Track شماره‌ی صفر شروع می‌کند.

- فرض کنید در ابتدا Head روی Track شماره‌ی پنجاه باشد و به سمت Track شماره‌ی صد در حرکت باشد؛ درخواست‌های جدول زیر را در نظر بگیرید.

Request	1	2	3	4	5	6	7	8	9	10
Track #	45	70	56	10	22	90	33	85	80	75

- در شکل زیر، ترتیب فرستاده شدن درخواست‌ها به دیسک (مکان Head) نشان داده شده است.



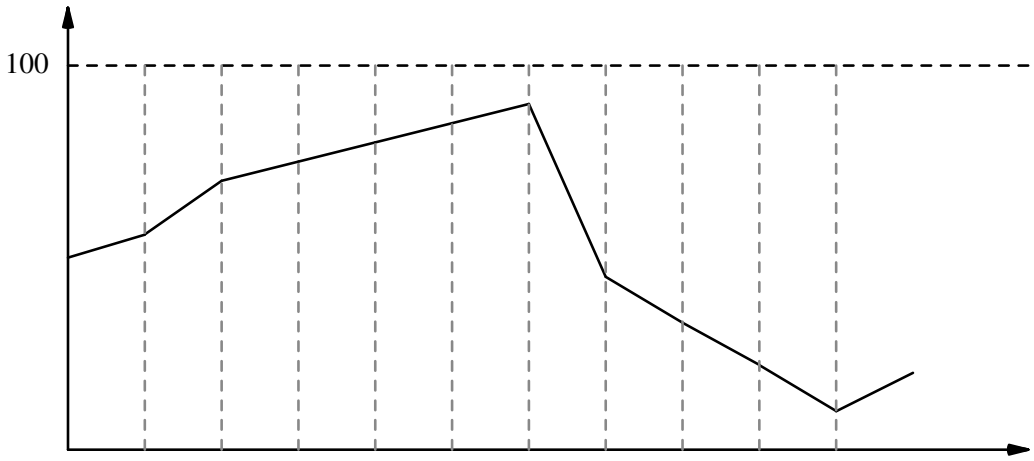
الگوریتم LOOK

در الگوریتم LOOK مشابه الگوریتم SCAN است با این تفاوت که Head فقط تا آخرین درخواست جلو می‌رود و از آنجا بر می‌گردد.

- فرض کنید در ابتدا Head روی Track شماره‌ی پنجاه باشد و به سمت Track شماره‌ی صد در حرکت باشد؛ درخواست‌های جدول زیر را در نظر بگیرید.

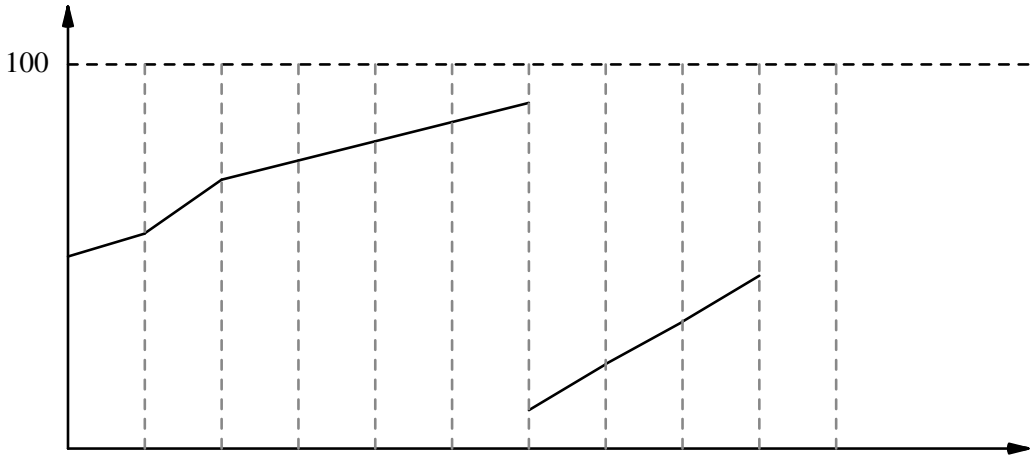
Request	1	2	3	4	5	6	7	8	9	10
Track #	45	70	56	10	22	90	33	85	80	75

- در شکل زیر، ترتیب فرستاده شدن درخواست‌ها به دیسک (مکان Head) نشان داده شده است.



الگوریتم C-LOOK

الگوریتم C-LOOK (Circular LOOK) مشابه الگوریتم LOOK هست با این تفاوت که وقتی سر خواندن به آخرین درخواست Track رسید حرکتش را از اولین درخواست بدون پاسخ (با کمترین شماره‌ی Track) به سمت بالا انجام می‌دهد.



- با پیشرفت تکنولوژی حجم دیسک‌ها بیشتر می‌شود و اندازه‌ی آنها کوچک‌تر.
- در نتیجه امروزه این امکان وجود دارد که چند دیسک را به یک کامپیوتر وصل کرد.

استفاده‌ی همزمان از چند دیسک با ظرفیت و سرعت متوسط به جای یک دیسک با ظرفیت بالا مزیت‌هایی دارد.

- Mirroring: چون می‌توان از داده‌ها چند نسخه روی دیسک‌های مختلف نگهداری کرد، با از کار افتادن یکی از دیسک‌ها همچنان می‌توان به اطلاعات دسترسی داشت و در نتیجه قابلیت اطمینان افزایش می‌یابد.
- Data-Stripping: چون به صورت همزمان می‌توان از چند دیسک مستقل خواند (یا نوشت) سرعت دسترسی به اطلاعات با وجود چند دیسک می‌تواند بیشتر از وقتی باشد که یک دیسک برای نگهداری اطلاعات استفاده شود.

در واقع حافظه‌ی RAID (Redundant Arrays of Independent Disks) از همین ایده استفاده می‌کنند.

- هر حافظه‌ی RAID از چند دیسک مستقل تشکیل می‌شود؛ به صورت سخت‌افزاری یا نرم‌افزاری شیوه‌ی قرار گرفتن بلوک‌های داده در دیسک‌ها مدیریت می‌شود.

تکنیک Data-Stripping به دو شکل انجام می‌شود.

- Block-Level Stripping: اگر k دیسک داشته باشیم، بلوک i-ام به دیسک $(i \bmod k) + ۱$ نوشته می‌شود.
- Bit-Level Stripping: اگر هشت دیسک داشته باشیم، بیت i-ام به دیسک i-ام نوشته می‌شود.

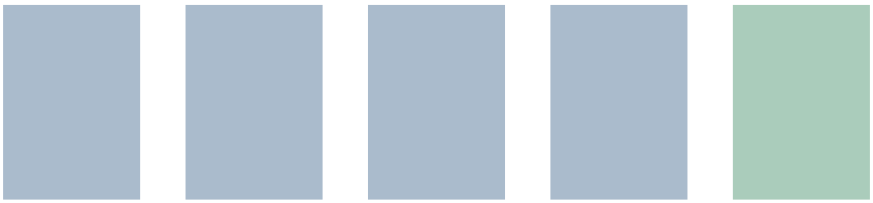
- RAID 0: Block-Level Stripping.

- RAID 1: Mirroring.

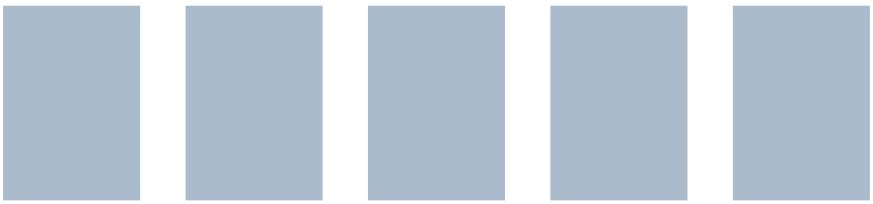
- RAID 2: Error-Correcting Codes (ECC). مشابه تصحیح خطا در حافظه.

- RAID 3: Bit-Interleaved Parity+ Bit-Level Stripping. دیسک می‌تواند تشخیص دهد که یک بلوک با خطا خوانده شده است.

- RAID 4: Block-Interleaved Parity + Block-Level Stripping.



- RAID 5: Block-Interleaved Distributed Parity + Block-Level Stripping.



- RAID 6: P + Q Redundancy Scheme. مشابه رده‌ی 5 با این تفاوت که به جای زوجیت، از روش‌های تصحیح خطایی مثل Reed-Solomon استفاده می‌کند.

- RAID 0+1: RAID 0 + RAID 1. ابتدا با RAID 0 عمل Stripping انجام می‌شود و سپس با RAID 1 از دیسک‌ها عمل Mirroring انجام می‌شود.

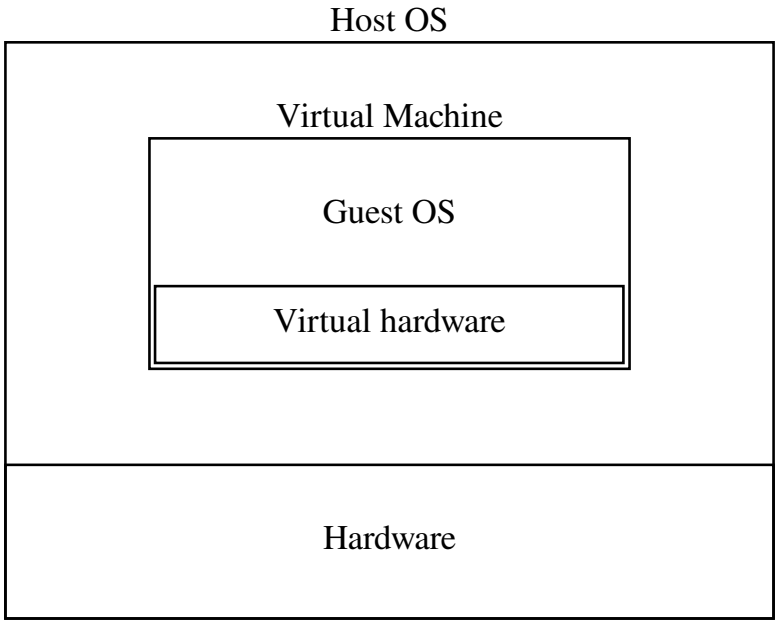
- RAID 1+0: RAID 1 + RAID 0. ابتدا با RAID 1 عمل Mirroring انجام می‌شود و سپس با RAID 0 از دیسک‌ها عمل Stripping انجام می‌شود.

- زمانبندی ورودی یا خروجی (I/O Scheduling): تعیین ترتیب اجرای درخواست‌های ورودی یا خروجی.
- Buffering: نگهداری موقتی داده‌هایی که ارسال می‌شوند.
- Caching: نگهداری اطلاعاتی که خوانده شده‌اند در حافظه برای افزایش سرعت دسترسی.
- دسترسی همزمان به دستگاه‌های ورودی و خروجی که باید به صورت انحصار متقابل استفاده شوند:

  الف) رزرو کردن دستگاه ورودی و خروجی.

  ب) ریختن اطلاعات ورودی و خروجی در فایلی به نام Spool و فرستادن آنها به دستگاه در آینده.

- با مفهوم Virtual Machine Manager یا Hypervisor آشنا شده‌ایم.



انواع مجازی‌سازی

- نوع صفر (Type 0 Hypervisors): مجازی‌سازی توسط سخت‌افزار انجام می‌شود.
- نوع یک: مجازی‌سازی توسط قسمتی نرم‌افزاری مثل سیستم‌عامل انجام می‌شود.
- نوع دو: مجازی‌سازی در پردازه‌ای در یک سیستم عامل انجام می‌شود.
- Paravirtualization: سیستم عامل میزبان تغییر داده می‌شود تا درخواست‌ها را به میزبان بفرستد.

- محافظت ماشین‌های مجازی و سیستم عامل میزبان.
- امکان معلق کردن (Suspend) ماشین مجازی.
- گرفتن Snapshot و گرفتن یک کپی از ماشین مجازی.
- بدون وقفه در سیستم عامل میزبان برای توسعه‌ی سیستم عامل.
- استفاده از Template برای ساختن ماشین‌های مجازی و کاهش هزینه‌ی مدیریت؛ نرم‌افزارهای آماده.
- امکان Live Migration.
- تغییر منابع با توجه به نیاز.