

فصل هشتم

مدیریت فایل و زمان‌بندی دیسک‌ها

۱-۱-۸ مقدمه

یکی از مهم‌ترین سرویس‌هایی که یک سیستم عامل ارایه می‌کند، مدیریت فایل است. یک فایل مجموعه‌ای از اطلاعات وابسته تعریف شده به وسیله سازنده فایل است.

فایل‌ها توسط سیستم عامل بر روی دستگاه‌های فیزیکی نگاشته می‌شوند. به منظور سهولت کار با فایل‌ها معمولاً آن‌ها را در فهرست‌هایی سازمان می‌دهند.

سیستم عامل در ارتباط با مدیریت فایل مسؤول فعالیت‌های زیر است:

- ۱- ایجاد و حذف فایل‌ها و فهرست‌ها.
- ۲- مهیا کردن ابزاری برای کار با فایل‌ها و فهرست‌ها.
- ۳- نگاشت فایل‌ها روی حافظه جانبی.
- ۴- ایجاد پشتیبانی از فایل‌ها بر روی حافظه ماندنی.
- ۵- مکانیسم حفاظتی برای کنترل دست‌یابی کاربران مختلف به فایل‌ها.

۱-۲-۸ سازمان سیستم عامل

از آنجایی که سیستم کامپیوتری باید برای استفاده مناسب و راحت باشد، سیستم عامل یک دیدگاه منطقی یکنواخت، از ذخیره‌سازی اطلاعات را تدارک می‌بینند. سیستم عامل ویژگی‌های فیزیکی دستگاه‌های ذخیره‌سازی اش را به صورت یک واحد ذخیره‌سازی منطقی به نام فایل تبدیل می‌نماید، به عبارتی دیگر فایل‌ها به وسیله سیستم عامل بر روی دستگاه‌های فیزیکی نگاشته می‌شوند. بنابرین سیستم فایل متشكل از دو قسمت مختلف است :

- ۱- مجموعه‌ای از فایل‌های حقیقی که هر یک متشكل از اطلاعات مرتبط هستند.
- ۲- ساختار فهرست‌ها که اطلاعاتی درباره تمام فایل‌ها در سیستم را در بر می‌گیرند.

۳-۸- ساختار فهرست

فایل‌ها در یک سیستم کامپیوتری به وسیله درایه‌هایی در فهرست دستگاه نشان داده می‌شوند. فهرست دستگاه اطلاعات تمام فایل‌های روی آن دستگاه را ثبت می‌نماید.

اطلاعات خاص هر فایل که در فهرست نگه داشته می‌شود، از یک سیستم عامل به سیستم عامل دیگری متغیر است. در هر درایو فهرست، می‌توان اطلاعات زیر را ثبت کرد. اگر چه تمام سیستم‌ها لزوماً دارای چنین اطلاعاتی نیستند.

۱- اسم فایل

۲- نوع فایل

۳- مکان فایل

۴- اندازه فایل

۵- موقعیت فعلی

۶- حفاظت

۷- میزان استفاده

۸- زمان، تاریخ و معرف فرآیند.

۴-۸- عملیات فایلی

فایل یک نوع داده انتزاعی است. برای تعریف مناسب یک فایل لازم است عملیاتی را که می‌توانند روی فایل اجرا گردند را مد نظر قرار داد. سیستم عامل عملیات ایجاد یک فایل، نوشتن یک فایل، خواندن یک فایل، بازگرداندن فایل و حذف فایل را مهیا می‌سازد.

۵-۸- شیوه‌های دست‌یابی

فایل اطلاعات را ذخیره می‌کند. هنگامی که فایلی مورد استفاده قرار می‌گیرد. این اطلاعات باید در دسترس باشند و به حافظه کامپیوتر خوانده شوند. روش‌های چندی وجود دارد که در طی آن‌ها اطلاعات در فایل‌ها می‌تواند مورد دست‌یابی قرار گیرند. برخی سیستم‌ها تنها یک شیوه دست‌یابی را تدارک می‌بینند. در سیستم‌های دیگر شیوه‌هایی دیگر شیوه‌هایی چندی وجود دارد و انتخاب شیوه مناسب برای یک کاربرد خاص یکی از مسایل مهم طراحی است.

۱- دسترسی ترتیبی: اطلاعات در فایل به ترتیب پردازش می‌شوند، به عبارتی دیگر یک رکورد بعد از دیگری مورد پردازش قرار می‌گیرد.

۲- دسترسی مستقیم: فایل به عنوان دنباله شماره‌گذاری شده از بلوک‌ها یا رکوردها دیده می‌شوند. یک دسترسی مستقیم به فایل امکان می‌دهد بلوک‌های دلخواه خوانده و یا نوشته شوند.

۳- دسترسی از طریق شاخص (Index Sequential): شاخصی برای فایل ایجاد می‌کنیم که این شاخص مانند شاخص انتهای یک کتاب، اشاره‌گرهایی را به بلوک‌های مختلف در بر می‌گیرد. برای یافتن یک ورودی به فایل ابتدا شاخص جستجو می‌گردد و سپس اشاره‌گری برای دست‌یابی مستقیم به فایل و ورودی مطلوب به کار می‌رود.

۶-۸- سازماندهی ساختار فهرست

فهرست اصولاً یک جدول نشانه است. سیستم عامل اسم نمادی فایل را می‌گیرد و فایل نامبرده را می‌یابد. شیوه‌های متفاوتی برای تعریف ساختار منطقی یک سیستم فهرست مورد استفاده قرار می‌گیرد.

فهرست تک سطحی: ساده‌ترین ساختار فهرست، فهرست تک سطحی است. کلیه فایل‌ها در یک فهرست یکسان قرار می‌گیرند. هنگامی که تعداد فایل‌ها افزایش می‌یابد یازمانی که بیشتر از یک کاربر موجود است. این فهرست محدودیت‌های قابل توجهی ایجاد می‌نماید.

فهرست دو سطحی: در این ساختار، هر کار بر دارای فهرست فایل خاص خودش (UFD) می‌باشد. همه UFD‌ها بر دارای ساختار مشابه هستند، اما تنها فایل‌های یک کار بر واحد را لیست می‌نماید. هنگامی که کاربری کاری را آغاز کند فهرست فایل اصلی سیستم (MFD) جستجو می‌شود. این فهرست بر حسب اسمی کاربران یا شماره منحصر به فردی شاخص گذاری شده و هر مدخل به UFD کاربر مربوطه اشاره می‌نماید. به طور کلی فهرست دو سطحی را می‌توان به عنوان یک درختواره‌ای با عمق ۲ در نظر گرفت.

فهرست‌های با ساختار درختواره‌ای با یک عمق دلخواه:

این ساختار امکان می‌دهد کاربران زیر فهرست‌های خاص خود را ایجاد نمایند. و بطبق آن‌ها فایل‌های خود را شکل و سازمان دهند. این درختواره دارای یک فهرست ریشه است. هر فایل در سیستم دارای مسیر منحصر به فرد خود می‌باشد. مسیر شامل مسیر حرکت از ریشه در تمام زیر فهرست‌ها تا فایل مورد نظر می‌باشد.

۷-۸- حفاظت فایل

هنگامی که اطلاعات در یک سیستم کامپیوتری نگهداری می‌گردند، مساله حفاظت از آن‌ها در برابر خرابی فیزیکی و دست‌یابی غیرمجاز داری اهمیت خواهد بود.

حفاظت از دست‌یابی‌های غیرمجاز از طرق مختلف امکان پذیر است. در برخی از سیستم‌ها اصولاً اجازه دست‌یابی به فایل‌ها به سایر کاربران داده نخواهد شد. انواع مختلفی از عملیات وجود دارند که می‌توانند به صورت کنترل شده اجرا گردند.

۱. خواندن از فایل

۲. نوشتن یا بازنویسی دیگر فایل.

۳. اجرای فایلی که در حافظه بار شده است.

۴. الحاق اطلاعات جدید به انتهای فایل.

۵. حذف فایل و آزاد ساختن فضای آن

برخی از شیوه‌های حفاظت فایل عبارتند از:

۱- نام‌گذاری: چنان‌چه کاربری نتواند فایلی را بنامد، قادر به دست‌یابی به آن نیست.

فرض بر این است که روشی برای به دست آوردن اسمی فایل‌های کاربران دیگر وجود ندارد.

۲- کلمات عبور: در این شیوه به هر فایل یک کلمه عبور تعلق می‌گیرد. و از طریق این کلمه مجوز دست‌یابی به هر فایل کنترل می‌گردد.

۳- لیست‌های دست‌یابی: یک لیست دست‌یابی به هر یک از فایل‌ها و فهرست‌ها اختصاص می‌یابد که نام کاربر و نوع دست‌یابی خاص آن کاربر در آن معین می‌گردد.

۴- گروه‌های دست‌یابی: مشکل اصلی درباره لیست‌های دست‌یابی طول آنهاست، با استفاده از یک لیست دست‌یابی خلاصه شده این مشکل را برطرف می‌کنیم.

به این ترتیب که سه گروه کاربری در ارتباط با هر فایل به شکل ذیل در نظر می‌گیریم.

الف - مالک: کسی که فایل را ایجاد می‌کند.

ب - گروه: مجموعه‌ای از کاربران که به صورت اشتراکی از فایل استفاده می‌نمایند.

ج - سایر کاربران

با این دسته بندی، فقط ۳ فیلد برای عمل حفاظت مورد نیاز خواهد بود، هر فیلد از تعدادی بیت تشکیل شده است که هر یک از این بیتها یک نوع دست‌یابی را کنترل می‌نمایند.

۸-۸- مدیریت فضای آزاد دیسک

از آنجا که میزان فضای آزاد دیسک محدود است، لازم است فضای آزاد حاصل از فایل‌های حذف شده برای فایل‌های جدید مجدد استفاده گردد. برای ثبت وضعیت فضای آزاد دیسک، سیستم نیاز به لیست فضای آزاد دارد.

۱-۸-۸ - بردار بیتی (Bitmap)

هر بلوک آزاد به وسیله یک بیت صفر نشان داده می‌شود و در غیراین صورت بیت متناظرش ۱ خواهد بود. مزیت اصلی آن، سادگی این روش است و برای پیدا کردن n بلوک آزاد متوالی بروی دیسک کارا می‌باشد. نگهداری این بردار در حافظه اصلی برای دیسک‌های کوچک ممکن است.

۲-۸-۸ - فهرست‌های پیوندی (Linked List)

در این روش تمام بلوک‌های آزاد دیسک به یکدیگر پیوند داده می‌شوند و اشاره گری به اولین بلوک آزاد، مقداردهی می‌گردد. این شیوه چندان کارا نمی‌باشد، زیرا برای پیمایش فهرست، باید هر بلوک آزاد خوانده شود که زمان ورودی / خروجی قابل ملاحظه‌ای می‌طلبد.

۳-۸-۸ - گروه بندی (Grouping)

آدرس‌های n بلوک آزاد را در اولین بلوک آزاد ذخیره می‌نمایند. آخرین آدرس، آدرسی است که بلوک دیگر را که حاوی n آدرس بلوک‌های آزاد دیگر است، نشان می‌دهد. اهمیت این روش این است که آدرس‌های تعداد زیادی از بلوک‌های آزاد را می‌توان به سرعت پیدا کرد.

۴-۸-۸ - شمارشی (Counting)

آدرس اولین بلوک آزاد و تعداد n بلوک آزاد متوالی، که بعد از اولین بلوک آزاد قرار گرفته‌اند، را ذخیره می‌نماییم. بدین ترتیب هر درایو در لیست فضای آزاد شامل آدرس یک بلوک و تعدادی بلوک آزاد متوالی بعدی خواهد بود. بدین ترتیب لیست فضای آزاد کوتاه تر خواهد شد.

۸-۹- روش‌های اختصاص فضای آزاد

از آنجا که تعداد زیادی فایل بر روی دیسک ذخیره می‌شود. مساله اصلی چگونگی اختصاص فضای آزاد به این فایل‌هاست، به طوری که فضای آزاد دیسک به طور موثری استفاده شود و فایل‌ها با سرعت زیاد مورد دست‌یابی قرار گیرند. فرض بر این است که فایل به صورت بلوک‌های متوالی دیده می‌شود و تمام کارهای ورودی / خروجی اصلی بر حسب بلوک‌ها صورت می‌پذیرد.

۸-۹-۱- تخصیص به شیوه پیوسته

تخصیص پیوسته فایل با آدرس اولین بلوک و طول فایل (بر حسب واحد بلوک) تعریف می‌گردد. بدین ترتیب درایه فهرست برای هر فایل، آدرس بلوک آغازین و طول فضای اختصاص یافته شده برای فایل را شامل خواهد شد.

معایب این روش تکه تکه شدن خارجی (External Fragmentation) و چگونگی تعیین فضای مورد نیاز برای یک فایل می‌باشد.

۸-۹-۲- تخصیص به شیوه پیوندی

در این روش هر فایل متشکل از یک لیست پیوندی از بلوک‌ها است. این بلوک‌ها می‌توانند در هر جای دیسک، پخش شده باشند. فهرست شامل اشاره گری به بلوک‌های اول و آخر فایل می‌باشد. اهمیت این روش آن است که مشکل تکه تکه شدن خارجی را حذف می‌نماید و نیازی به مشخص شدن اندازه فایل در زمان ایجاد آن نخواهد بود. مشکل اصلی آن است که فقط در دست‌یابی ترتیبی فایل‌ها به طور موثر عمل می‌نماید. عیب دیگر این شیوه مقدار حافظه مصرف شده برای اشاره‌گرها در بلوک‌هاست و از قابلیت اطمینان کمی نیز برخوردار است زیرا اشاره‌گرها در داخل بلوک‌ها در سرتاسر دیسک پخش می‌باشد.

۸-۹-۳- تخصیص شاخص دار

هر فایل دارای بلوکی بنام بلوک شاخص می‌باشد که متشکل از تمام اشاره‌گرهایی است که به بلوک‌های آن فایل اشاره می‌نماید. بدین معنی که بلوک شاخص آرایه‌ای از آدرس‌های بلوک‌های آن فایل ایجاد می‌نماید. فهرست، شامل آدرس بلوک شاخص خواهد بود. این روش امکان دسترسی مستقیم را در اختیار قرار می‌دهد و ضمناً تکه تکه شدن خارجی نیز پیش نخواهد آمد. مشکل این شیوه فضای هرزی است که به ازای بلوک شاخص مصرف می‌شود.

۸-۹-۴- شیوه ترکیبی

این روش در سیستم BSD-UNIX استفاده می‌شود و به این ترتیب است که تعدادی اشاره گر در بلوک شاخص، در فهرست دستگاه ذخیره می‌شود. این اشاره‌گرها به ۴ شکل مختلف ظاهر می‌شوند:

۱- اشاره‌گرهایی که به بلوک‌های مستقیم اشاره می‌نمایند.

۲- اشاره‌گرهایی که به بلوک غیرمستقیم واحد (Single Indirect Block) اشاره می‌نمایند.

۳- اشاره‌گرهایی که به بلوک غیرمستقیم مضاعف (Double Indirect Block) اشاره می‌نمایند.

۴- فقط یک اشاره گر نوع آخر به یک بلوک غیرمستقیم سه گانه (Triple Indirect Block) اشاره می‌نماید.

بلوک غیرمستقیم واحد یک بلوک شاخص است که شامل داده‌ای نمی‌باشد و در عوض آدرس بلوک‌هایی را در خود حفظ می‌کند که حاوی داده‌ها هستند.

بلوک غیرمستقیم مضاعف یک بلوک شاخص است که آدرس غیرمستقیم واحد را در خود حفظ می‌نماید. بلوک غیرمستقیم سه گانه یک بلوک شاخص است که آدرس بلوک‌های غیرمستقیم مضاعف را در خود حفظ می‌نماید.

۸-۱۰-۸ - پارامترهای کارایی دیسک

هنگامی که فرآیندی یک درخواست ورودی / خروجی می‌دهد، ابتدا باید برای در دسترس بودن دستگاه در یک صفحه قرار گیرد. پس از فراهم شدن، دستگاه به فرآیند مزبور تخصیص می‌یابد. پس از آن، برای دستیابی به رکوردهای خاص از داده‌ها در روی یک دستگاه با نوک متحرک (Movable Head)، عملیات ذیل باید انجام گیرد:

۱- ابتدا نوک خواندن - نوشتن باید بر روی شیار (استوانه مورد نظر) مورد درخواست قرار گیرد. زمان لازم برای قرار گرفتن نوک بر روی شیار را زمان پیگرد (Seek Time) می‌نامند.

۲- پس از انتخاب شیار، صفحه دیسک طوری چرخش می‌کند که ابتدای قطاع در مقابل نوک قرار می‌گیرد. زمان لازم برای رسیدن قطاع مورد نظر به نوک را تاخیر چرخشی (Rotational Time) می‌نامند.

۳- رکورد که دارای اندازه دلخواهی است (حداکثر به اندازه خودشیار) باید به وسیله نوک پیموده شود. زمان لازم برای این پیمایش را زمان انتقال (Transmission Time) می‌نامند. متوسط کل زمان دسترسی را به صورت ذیل در نظر می‌گیریم:

$$T_{access} = T_{seek} + (1/2r) + (b/rN)$$

T_{seek} = متوسط زمان پیگرد

r = سرعت چرخش بر حسب دور بر ثانیه

b = تعداد بایتها که باید منتقل شود

N = تعداد بایتها یک شیار

۸-۱۱-۸ - الگوریتم‌های زمان‌بندی دیسک

هنگامی که فرآیندی درخواست ورودی/خروجی می‌دهد، اگر گرداننده دیسک (Disk Drive) و کنترل کننده مورد نظر در دسترس باشد، درخواست بلاfaciale بهترتبی که در بخش قبل توضیح داده شد سرویس داده می‌شود، در غیر این صورت هر درخواست جدیدی برای سرویس، لازم است در صفحه درخواست‌هایی به تعویق افتاده برای آن وسیله قرار داده شود. وقتی یک تقاضا کامل می‌شود، سیستم عامل این فرصت را دارد که یک درخواست معوقه را برای سرویس برگزیند. در ذیل به الگوریتم‌های زمان‌بندی دیسک برای یک انتخاب مناسب از مجموعه درخواست‌ها برای ورودی / خروجی می‌پردازیم.

۱۱-۸-۱ - الگوریتم FCFS

در این الگوریتم به اولین درخواستی که از راه برسد، سرویس داده می‌شود. تقاضای با اولویت بالا تاثیری در ترتیب ایجاد شده در زمان‌بندی نخواهد داشت.

۱۱-۸-۲ - الگوریتم کوتاه ترین زمان جست وجو (SSTF)

درخواستی که با حداقل مسافت جستجو پاسخ داده می‌شود، در ابتدا سرویس می‌گیرد. به عبارت دیگر تقاضا، با حداقل زمان جست وجو برآورده خواهد شد. هنگامی که حجم کار متعادلی داشته باشیم، دارای توان عملیات بهتری نسبت به FCFS خواهد بود. این الگوریتم می‌تواند سبب قحطی زدگی بعضی از درخواست‌ها گردد.

۱۱-۸-۳ - الگوریتم پویش (SCAN)

بازوی دیسک در یک جهت حرکت کرده و سر راه، تمام درخواست‌های منتظر را برآورده می‌کند تا جایی که به آخرین شیار برسد یا درخواست دیگری در آن جهت وجود نداشته باشد (به نام سیاست LOOK موسوم است). سپس جهت معکوس می‌شود و مرور در جهت دیگر ادامه می‌یابد و مجدداً درخواست به ترتیب رسیدگی می‌شود. این الگوریتم بنام الگوریتم آسانسور هم نامیده می‌شود.

۴-۱۱-۸ - الگوریتم پویش چرخشی (Circular Scan)

این سیاست در یک جهت پویش می‌نماید. یعنی هنگامی که به آخرین شیار در یک جهت رسید، بازو به انتهای دیگر دیسک باز می‌گردد و مرور دوباره آغاز می‌شود. این سیاست حداقل تاخیر درخواست‌های جدید را محدود می‌سازد.

۴-۱۱-۹ - الگوریتم پویش N گامی و پویش سریع (FSCAN, N-STEP-SCAN)

سیاست پویش N گامی، صفت درخواست دیسک را به زیر‌صفهایی به طول N می‌شکند. هنگامی که صفحی در حال پردازش است، درخواست‌های جدید باید به انتهای صفحه درخواست اضافه شوند.

پویش سریع سیاستی است که دو زیر‌صف را به کار می‌گیرد. هنگامی که یکی از صفحه‌ها آغاز می‌شود، تمامی درخواست‌ها در صفحه دیگر قرار می‌گیرد (در طی پویش تمام درخواست‌های جدید در صفحه دیگر قرار می‌گیرند). بنابراین رسیدگی به درخواست‌های جدید تا زمان پردازش کامل درخواست‌های قدیمی به تعویق می‌افتد.

تست‌های مربوط به مبحث زمانبندی دیسک‌ها

۱- یک دستگاه دیسک خوان با استفاده از روش آسانسور سیلندرها را جستجو کرده و عمل خواندن را انجام می‌دهد.
اگر تقاضاهایی به ترتیب برای سیلندرهای ۳۸و۴۰و۲۰و۲۲و۱۰ روی سیلندر ۲۰ به طرف
بالا در حرکت باشد و عمیلی یک سیلندر به سیلندر بعدی برود کل زمان جستجو برای سیلندرها
چقدر است؟

الف) ٣٤٨ ميلي ثانية ب) ٣٦٠ ميلي ثانية ج) ٨٧٦ ميلي ثانية د) ٨٩٢ ميلي ثانية

کل زمانی که جهت انجام این درخواست‌ها صرف حرکت بازو می‌شود برابر است با:

درخواست‌های رسیده به ترتیب مقابل (چپ به راست) باشد: ۲۳۱ و ۲۵۶ و ۲۴۵ و ۱۳۳ و ۲۸۳

مورد استفاده آگر زمان حرکت بازو برای هر سیلندر ۳ msec می‌باشد. اگر سیلندر ۲۵۰ وجهت حرکت آن به طرف بالا (سیلندرهای با شماره بزرگ‌تر) و سیاست ۲ - موقعیت بازو در یک دیسک گردان سیلندرهای با شماره بزرگ‌تر) و سیاست

الف) ٩٥١ msec (ج) ١٢٨٣ msec (د) ٧٢٩ msec (ب) ٥٤٩ msec

۳- یک دستگاه دیسک خوان با استفاده از روش Look C-Scan سیلندرها را جستجو کرده و عمل خواندن را انجام می‌دهد.
اگر تقاضاهایی به ترتیب برای سیلندرهای ۱۰۶ و ۱۰۴ و ۱۰۲ و ۱۰۳ به طرف HEAD دستگاه روی سیلندر ۲۰ داده شود و
بالا در حرکت باشد و ۶۰ میلی ثانیه طول بکشد تا HEAD از یک سیلندر به سیلندر بعدی برود کل زمان جستجو برای سیلندرها
چقدر است؟

الف) ٣٦٠ ميلي ثانية ب) ٨٧٦ ميلي ثانية ج) ٣٩٦ ميلي ثانية د) ٣٤٨٠ ميلي ثانية

۴- در سیستم حافظه اصلی حداکثر برازی دو فرآیند کاربر و سیستم عامل جا دارد. فرآیندها هم اندازه می‌باشند، به این ترتیب فرآیندها باید به داخل و خارج حافظه جابه‌جا شوند. اطلاعات زیر در مورد دیسک موجود سیستم به قرار ذیل است.

متوجه زمان حرکت هد = ۳۰ میلی ثانیه
حداقل زمان حرکت هد = ۱۰ میلی ثانیه
سرعت چرخش دیسک = ۱۵ میلی ثانیه در هر دور
گنجایش هر شیار = ۱ کیلو بایت
اندازه فرآیند = ۳۵ کیلو بایت (شیارهای مجاور)

با فرض این که طول هربرش زمانی ۱۰۰ میلی ثانیه حداقل بیش از ۳ ثانیه باشد، این سیستم برای تضمین زمان پاسخ سرویس دهد.

الف) ٧ نفر ب) ٨ نفر ج) ١٠ نفر د) ٣٠ نفر

۵- یک فایل حاوی ۱۴ رکورد به ترتیب در شیارهای ۷,۵,۳,۱ ذخیره شده‌اند. نوک دیسک بر روی شیار ۴ قطاع صفر می‌باشد که به طرف شیارهای با شماره کوچک‌تر در حرکت آن‌ها از صفر در جهت عقربه ساعت تا شماره ۳ و دیسک نیز در همان جهت می‌چرخد. فرض کنید در خواستهای ذیل برای خواندن رکورد شماره ۱ بده ترتیب بده سیستم رسیده‌اند و شماره اولین روش آسانسوری استفاده کند (SCAN) کل زمان معرف شده برای انجام درخواست‌ها چقدر خواهد بود؟

در خروجی است $\{7, 1, 12\}$

زمان لازم برای حرکت نوک از پیک شیار به شیار متواالی 10 میلی ثانیه زمان یک دور چرخش دیسک 20 میلی ثانیه در نظر بگیرید.

۶ - یکی از مسؤولیت‌های سیستم‌های عامل زمان‌بندی دیسک می‌باشد. سیاست‌های مختلفی برای زمان‌بندی دیسک وجود دارد که بعضی از آن‌ها ممکن است منجر به قحطی زدگی گردد. کدام یک از گزینه‌های زیر در مورد سیاست‌های زمان‌بندی ممکن است منجر به قحطی زدگی گردد؟

(الف) Last in Firstout, SSTF, Priority
Look, SSTF, SCAN (ب) Last in Firstout, SSTF, Priority

(ج) Priority, Look, CSCAN (د) SSTF, CSCAN, Last in First out

۷ - موقعیت اولیه در یک دیسک گردان، سیلندر 25 و سیاست مورد استفاده آسانسور با جهت اولیه رو به پایین می‌باشد. اگر زمان حرکت بازو از هر سیلندر مجاور 10 میلی ثانیه و درخواست‌ها به ترتیب زیر باشند (برای مثال، درخواست دوم 90 میلی ثانیه پس از درخواست اول برای خواندن یا نوشت 2 بلوک پشت سر هم از سیلندر شماره 40 صادر شده است) با فرض این‌که زمان یک دور چرخش دیسک برابر 20 میلی ثانیه است و هر شیار دارای 4 بلوک است، کل زمانی که جهت انجام این درخواست‌ها صرف می‌شود برابر است با:

شماره سیلندر	زمان درخواست	تعداد بلوک درخواست شده
35	0	2 بلوک پشت سر هم
40	90 ms	2 بلوک پشت سر هم
37	95 ms	2 بلوک پشت سر هم
25	195 ms	2 بلوک پشت سر هم
30	200 ms	1 بلوک

ب) 400 میلی ثانیه

الف) 405 میلی ثانیه

د) 385 میلی ثانیه

ج) 395 میلی ثانیه

۸ - یک دیسک سخت در کامپیوتر شخصی را با مشخصات ذیل در نظر بگیرید:

مگابایت در ثانیه = 48.25 = حداقل نرخ انتقال داخلی از دیسک به بافر

اگر یک فایل 200 مگابایتی در سکتورهای تصادفی دیسکی ذخیره شده باشد خواندن فایل چند ثانیه طول می‌کشد؟

سرعت دوران = 5400 دور در دقیقه و متوسط زمان جستجو = 12 میلی ثانیه
اندازه سکتور = 512 بایت

د) 170 دقیقه

ج) 100 دقیقه

ب) 150 دقیقه

الف) 120 دقیقه

جواب‌های تشریحی تست‌های مربوط به مبحث زمان‌بندی دیسک‌ها

- ۱ - بر اساس الگوریتم آسانسور، Head دیسک گردان از روی سیلندر ۲۰ به طرف بالا حرکت می‌نماید و تمام درخواست‌های مورد نیاز را در این جهت پاسخ می‌دهد و بعد از اتمام I/O بر روی سیلندر ۴۰ به طرف پایین حرکت کرده و باقیمانده سیلندرها تا کوچک‌ترین سیلندر مورد تقاضا پاسخ می‌دهد.

$20 \xrightarrow{2} 22 \xrightarrow{16} 38 \xrightarrow{2} 40 \xrightarrow{30} 10 \xrightarrow{4} 6 \xrightarrow{4} 2$

تعداد کل سیلندرهای طی شده = $58 = 2 + 16 + 2 + 30 + 4 + 4$

کل زمان طی شده = $58 \times 6 = 348$ میلی ثانیه

گزینه اول صحیح است.

- ۲ - بر اساس الگوریتم SSTF، به کوتاه‌ترین زمان جستجو (Seek) سرویس داده می‌شود. موقعیت فعلی بازوی دیسک گردان بر روی سیلندر ۲۵۰ می‌باشد، نزدیک‌ترین درخواست سیلندر ۲۴۵ و سپس به ترتیب ۲۴۳، ۲۳۱، ۲۵۶ و در آخر سیلندر ۱۳۳ است. به ترتیب ذیل فاصله سیلندرها محاسبه می‌گردد.

سیلندرهای طی شده

$250 \xrightarrow{5 \text{ cyl}} 245 \xrightarrow{11 \text{ cyl}} 256 \xrightarrow{25 \text{ cyl}} 231 \xrightarrow{52 \text{ cyl}} 283 \xrightarrow{150 \text{ cyl}} 133$

جمع کل بر روی سیلندرها حرکت بازو = ۲۴۳

کل زمان طی شده = $243 \times 3 = 729$

گزینه چهارم صحیح است.

- ۳ - بر اساس الگوریتم C-Scan Look با توجه به فرض مساله، به ترتیب ذیل از چپ به راست Head دیسک گردان حرکت می‌نماید و به درخواست‌های مورد نیاز پاسخ می‌دهد.

سیلندرهای طی شده

$20 \xrightarrow{2} 22 \xrightarrow{16} 38 \xrightarrow{2} 40 \xrightarrow{38} 2 \xrightarrow{4} 6 \xrightarrow{4} 10$

تعداد کل سیلندرهای طی شده = $66 = 2 + 16 + 2 + 38 + 4 + 4$

کل زمان طی شده = $66 \times 6 = 396$ میلی ثانیه

گزینه سوم صحیح است

- ۴ - چون فضای حافظه برای دو فرآیند جا دارد پس هنگامی که پردازنده به یک فرآیند سرویس می‌دهد در همان زمان می‌تواند فرآیند دوم را از حافظه خارج و فرآیند مورد درخواست را به داخل حافظه منتقل نماید، بنابراین مدت زمانی را که طول می‌کشد تا یک پردازش به خارج و پردازش بعدی به داخل انتقال یابد، به شکل ذیل محاسبه می‌نماییم:

Context Switch Time = Swapin Time + SwapOut Time

Swapin Time = SwapOut Time

سپس زمان انتقال را برای ۳۵ کیلوبایت محاسبه می‌نماییم.

$$\text{Swapin Time} = \left(30 + \frac{15}{2} + 15 \right) + 2 \left(10 + \frac{15}{2} + 15 \right) + \left(10 + \frac{15}{2} + \frac{15}{2} \right)$$

زمان انتقال ۱۰
کیلوبایت اول

زمان انتقال ۵
کیلوبایت باقیمانده

زمان انتقال ۵
کیلوبایت باقیمانده

$$\text{Swapin Time} = 52.5 + 65 + 25 = 142.5$$

$$\text{Context Switch Time} = 2 \times 142.5 = 285$$

بنابراین راه اندازی هر فرآیند برای اجرا حدود ۳ برش زمانی (۳۰۰ میلی ثانیه) به طول می‌انجامد، نتیجتاً این سیستم برای تضمین زمان پاسخ ۳ ثانیه (۳ هزار میلی ثانیه) حداکثر به ۱۰ پردازش سرویس می‌دهد (زیرا هر پردازش ۳۰۰ میلی ثانیه سرویس پردازنده را می‌تواند داشته باشد)

$$142.5 = 0.2 + 27.5 + 30 + 0.25 = 142.5$$

۵ - گزینه ۳ صحیح است.

رکوردهای فایل به ترتیب ذیل بر روی شیارها قرار گرفته‌اند.

$$T_1(1, 2, 3, 4), T_3(5, 6, 7, 8), T_5(9, 10, 11, 12)$$

$$T_7(13, 14)$$

حرکت نوک دیسک از شیار 4 به طرف پایین و سپس به طرف بالا به ترتیب ذیل می‌باشد.

(بر اساس الگوریتم SCAN)

$$T_4 \rightarrow T_3 \rightarrow T_1 \rightarrow T_5$$

نوک دیسک از T_4 به طرف T_3 (رکورد شماره 7) حرکت کرده. سپس به ترتیب به طرف T_1 (رکورد شماره 1) و در انتهای طرف T_5 (رکورد شماره 12) ادامه حرکت داده رکوردها را سرویس می‌دهد.

زمان خواندن رکورد 7 + زمان قرار گرفتن نوک دیسک روی رکورد 7 + زمان حرکت طی شده به شیار 3 = زمان سرویس دهی رکورد شماره 7
 $10^{ms} + 10 + 5 = 25$

زمان خواندن رکورد 1 + زمان قرار گرفتن نوک دیسک روی رکورد 1 + زمان حرکت طی شده از شیار 3 به شیار شماره 1 = زمان سرویس دهی به رکورد شماره 1
 $20 + 10 + 5 = 35$

زمان خواندن رکورد 12 + زمان قرار گرفتن نوک دیسک روی رکورد 12 + زمان حرکت طی شده از شیار شماره 1 به شیار شماره 5 = زمان سرویس دهی به رکورد شماره 12
 $50^{ms} + 10 + 5 = 65$

کل زمان سرویس دهی به تمام رکوردهای مورد درخواست

۶ - گزینه اول صحیح است.

۷ - گزینه سوم صحیح است.

در لحظه صفر فقط یک درخواست موجود است، بنابراین نوک خواندن و نوشتن دیسک از سیلندر 25 به سیلندر 35 منتقل می‌شود.

$$25 \longrightarrow 35$$

این انتقال 100 میلی ثانیه طول می‌کشد، پارامتر بعدی زمان تاخیر چرخشی است که معادل $\frac{20}{2} = 10$ می‌باشد و سپس پارامتر زمان انتقال محاسبه می‌شود که دو بلوك پشت سرهم باید خوانده شود، بنابراین 10 میلی ثانیه طول می‌کشد تا این دو بلوك خوانده شود

پس تا اینجا (100+10+10) میلی ثانیه طول می‌کشد تا درخواست اول جواب داده شود.

در این لحظه دو درخواست {40, 37} موجود است که بر اساس الگوریتم آسانسور اول به سیلندر 37 جواب می‌دهیم و سپس به سیلندر 40 زمان جستجوی سیلندر از 35 به 37، 20 میلی ثانیه طول می‌کشد.

زمان تاخیر چرخشی $\frac{20}{2}$ و زمان انتقال 2 بلوک پشت سرهم 10 میلی ثانیه که جمعاً برابر 40 میلی ثانیه می‌شود، به همین ترتیب کل زمانی که برای سیلندر 40 صرف می‌شود برابر $(30+10+10)$ میلی ثانیه خواهد شد.

دو درخواست بعدی هم به ترتیب اول سیلندر 30 و سپس سیلندر 25 را پاسخ می‌دهیم.

$$\text{کل زمان برای انجام درخواست 30} = \left(100 + \frac{20}{2} + 5 \right) = 115$$

$$\text{کل زمان برای انجام درخواست 25} = \left(50 + \frac{20}{2} + 10 \right) = 70$$

$$\text{کل زمان برای انجام درخواستها} = 120 + 40 + 50 + 115 + 70 = 395 \text{ میلی ثانیه}$$

- زمان انتقال + زمان تاخیر چرخشی + زمان جستجو = زمان خواندن

$$\text{میلی ثانیه} = \frac{60000}{5400} \times \frac{1}{2} = \text{زمان تاخیر چرخشی}$$

$$\text{میلی ثانیه} = \text{زمان جستجو}$$

$$\text{میلی ثانیه} = \frac{512}{48.25 \times 2^{20}} \times 1000 = \text{زمان انتقال}$$

$$= 17.6 \text{ میلی ثانیه} = \text{زمان خواندن یک سکتور}$$

$$\text{دقیقه} = 120 = \frac{2^{20}}{512} \times 7213 = \text{کل زمان خواندن}$$

گزینه اول صحیح است.