

سیستم‌های عامل
دکتر زرندی



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین سری دوم

۱۶ مهر ۱۴۰۳



سوال اول

به سوالات زیر در مورد وقفه‌ها پاسخ دهید.

۱. وقفه چیست؟ کلاس‌های مختلف وقفه را به صورت مختصر توضیح دهید.

پاسخ

وقفه، به رویداد یا سیگنالی گفته می‌شود که پردازنده را از یک کار جاری متوقف کرده و به پردازش یک کار فوری‌تر یا مهم‌تر هدایت می‌کند. این وقفه می‌تواند توسط سخت‌افزار یا نرم‌افزار ایجاد شود و به منظور مدیریت و پاسخگویی به رویدادهای مختلف در سیستم عامل استفاده می‌شود. به طور کلی می‌توان وقفه‌ها را به ۴ کلاس دسته‌بندی نمود:

(آ) برنامه: ایجاد شده توسط شرایطی که در نتیجه اجرای یک دستور رخ می‌دهد، مانند سرریز محاسباتی، تقسیم بر صفر، تلاش برای اجرای یک دستور العمل غیرمجاز، یا ارجاع به خارج از فضای مجاز حافظه کاربر.

(ب) تایمر: ایجاد شده توسط یک تایمر درون پردازنده. این وقفه به سیستم عامل امکان می‌دهد تا وظایف خاصی را به صورت منظم انجام دهد.

(ج) ورودی/خروجی: توسط کنترل‌کننده ورودی/خروجی ایجاد می‌شود، برای اعلام اتمام عادی یک عملیات یا اعلام انواع مختلفی از شرایط خطا.

۲. به هنگام وقوع وقفه، پردازنده چه اطلاعاتی را در پشته ذخیره می‌کند؟ دلیل استفاده از پشته چیست؟

پاسخ

هنگام وقوع وقفه، پردازنده باید اجرای برنامه فعلی را متوقف کرده و به پردازش رویداد وقفه بپردازد. برای اینکه بتواند پس از اتمام وقفه به اجرای برنامه اصلی بازگردد، نیاز دارد اطلاعاتی را که مربوط به وضعیت فعلی اجرای برنامه است، ذخیره کند. این اطلاعات شامل:

(آ) Program Counter: برای اینکه بدانیم از کجای برنامه اصلی به ISR جامپ زدیم

(ب) General Purpose Registers: برای اینکه مقادیر موجود در ثبات‌های عمومی که ممکن است در حین پردازش وقفه تغییر کنند، در پشته ذخیره می‌شوند تا بعد از اتمام وقفه، مقادیر اصلی بازیابی شوند.

(ج) Processor Status Word: که شامل اطلاعاتی درمورد پردازنده مانند Flag هاست.

(د) Stack Pointer: همچنین نیاز است آدرس آخرین مقدار ذخیره‌شده در پشته نیز ذخیره شود.

پاسخ

پشته یکی از ساده‌ترین و مؤثرترین ساختارهای داده برای مدیریت اطلاعات در زمان وقوع وقفه است. دلیل اصلی استفاده از پشته این است که آخرین دستور یا داده‌ای که ذخیره می‌شود، اولین داده‌ای است که باید بازیابی شود (Last In, First Out - LIFO). این ویژگی پشته باعث می‌شود پردازنده بتواند وضعیت برنامه را به درستی ذخیره کرده و پس از پایان وقفه به همان وضعیت بازگردد. استفاده از پشته همچنین امکان مدیریت خودکار و مرتب اطلاعات بدون نیاز به تخصیص دستی حافظه را فراهم می‌کند.

در ادامه تصویری از فرایند رسیدگی به وقفه‌ها از کتاب Stallings آورده شده است:

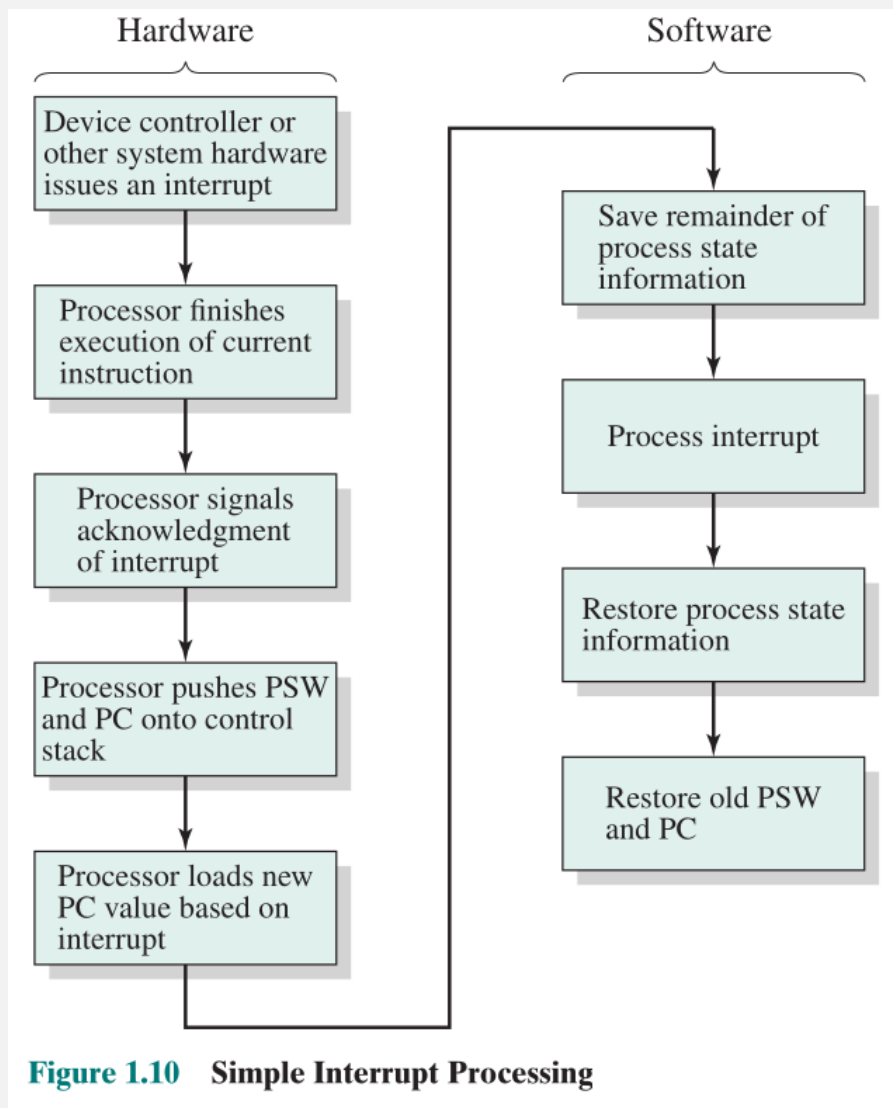


Figure 1.10 Simple Interrupt Processing

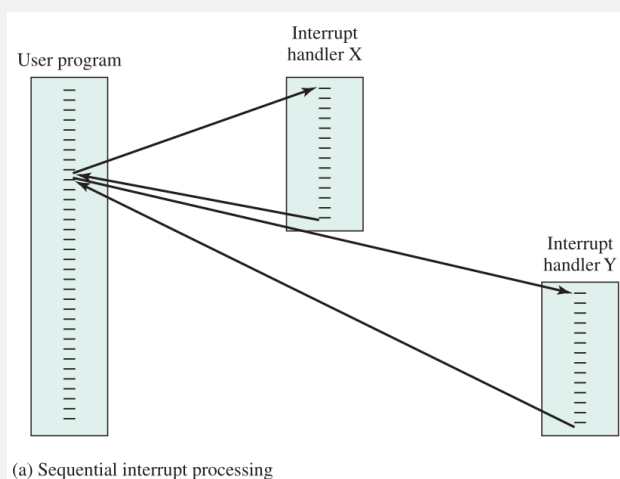
شکل ۱: فرایند رسیدگی به وقفه

۳. رویکردهای استفاده شده برای رسیدگی به وقفه‌های متعدد را بیان کنید و آنها را به صورت مختصر توضیح دهید.

پاسخ

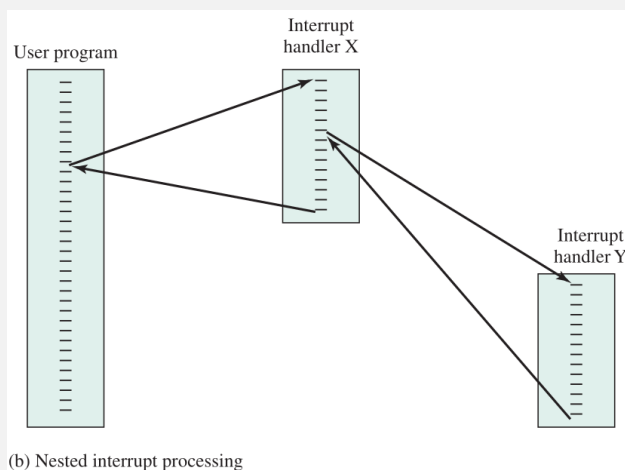
مطابق با کتاب آقای Stallings دو رویکرد برای مقابله با وقفه‌های متعدد وجود دارد.

(آ) اولین روش این است که در هنگام پردازش یک وقفه، تمامی وقفه‌های دیگر غیرفعال شوند. (وقفه غیرفعال به این معنی است که پردازنده هر سیگنال جدید درخواست وقفه را نادیده می‌گیرد). اگر در این مدت یک وقفه رخ دهد، معمولاً در حالت معلق باقی می‌ماند و بعد از اینکه پردازنده وقفه‌ها را دوباره فعال کرد، بررسی می‌شود. بنابراین، اگر وقفه‌ای هنگام اجرای برنامه‌ی کاربر رخ دهد، بلافاصله وقفه‌ها غیرفعال می‌شوند. پس از تکمیل ISR وقفه ایجاد شده، وقفه‌ها دوباره فعال می‌شوند و قبل از ادامه‌ی برنامه‌ی کاربر، پردازنده بررسی می‌کند که آیا وقفه‌های اضافی رخ داده‌اند یا خیر. (شکل زیر)



شکل ۲: رویکرد اول رسیدگی به وقفه‌ها

(ب) روش دوم این است که برای وقفه‌ها اولویت‌هایی تعریف شود و به یک وقفه با اولویت بالاتر اجازه داده شود تا روتین مدیریت وقفه با اولویت پایین‌تر را قطع کند. (شکل زیر)



شکل ۳: رویکرد دوم رسیدگی به وقفه‌ها

۴. پدیده سرریز پشته چه زمانی در وقفه‌های تو در تو رخ می‌دهد و برای حل این مشکل چه تدبیری اندیشیده شده است؟

پاسخ

در وقفه‌های تو در تو، پدیده Stack Overflow زمانی رخ می‌دهد که اطلاعات بیشتری نسبت به ظرفیت پشته در آن ذخیره شود. همانطور که در قسمت قبل بحث شد، در وقفه‌های تو در تو، ممکن است یک وقفه جدید در حین پردازش وقفه قبلی رخ دهد، و پردازنده مجبور می‌شود وضعیت جاری خود را (شامل ثبات‌ها و اطلاعات اجرایی) در پشته ذخیره کند. اگر تعداد وقفه‌ها به حدی زیاد شود که ظرفیت پشته پر شود، سرریز پشته اتفاق می‌افتد و این باعث از بین رفتن اطلاعات و خرابی سیستم می‌شود. دو مورد از تدابیر اندیشیده شده برای جلوگیری از این مشکل در قسمت قبل بیان شد. یعنی غیرفعال کردن وقفه‌های جدید به‌هنگام انجام ISR یک وقفه فعال و اولویت بندی وقفه‌ها. دو راهکار دیگر هم می‌توان پیشنهاد داد که یکی افزایش اندازه پشته است (البته در صورتی که بتوان این کار را انجام داد) و اگر نمی‌توانستیم اندازه پشته را تغییر دهیم می‌توان از یک حافظه جایگزین یا یک پشته مجزا در کنار پشته اصلی استفاده نمود.

سوال دوم

دو حالت اصلی^۱ عملیات‌ها در سیستم‌عامل را نام برده و هرکدام را به صورت مختصر توضیح دهید.

پاسخ

در OS دو حالت اصلی عملیات وجود دارد که به آن‌ها User Mode و Kernel Mode گفته می‌شود. در ادامه به توضیح مختصری از وظایف هریک می‌پردازیم:

۱. User Mode: در این حالت، برنامه‌های کاربردی یا نرم‌افزارهایی که توسط کاربر اجرا می‌شوند، عمل می‌کنند. در حالت کاربر، دسترسی مستقیم به منابع حیاتی سیستم مانند سخت‌افزار، حافظه یا تجهیزات ورودی/خروجی وجود ندارد. اگر برنامه‌ای در این حالت نیاز به دسترسی به منابع سیستم داشته باشد، باید از طریق System Call ها به حالت کرنل درخواست بدهد. این محدودیت‌ها برای جلوگیری از دسترسی مستقیم برنامه‌ها به سخت‌افزار و حفاظت از امنیت سیستم اعمال می‌شود.

۲. Kernel Mode: در این حالت، سیستم‌عامل به منابع حیاتی و مستقیم سخت‌افزار دسترسی کامل دارد و می‌تواند هرگونه عملیات لازم را اجرا کند. این حالت برای انجام وظایف مهم سیستم‌عامل مثل مدیریت حافظه، مدیریت پردازش‌ها، و کنترل سخت‌افزار استفاده می‌شود. در حالت کرنل، هیچ محدودیتی برای دسترسی به منابع وجود ندارد و دسترسی کامل به حافظه و دستگاه‌ها امکان‌پذیر است.

^۱ Mode

سوال سوم

نحوه عملکرد DMA را توضیح دهید. نحوه همکاری DMA و پردازنده به چه صورت است؟ پردازنده به چه صورت از به پایان رسیدن کار DMA مطلع می‌شود؟

پاسخ

وقتی قرار است حجم زیادی از داده‌ها منتقل شود، از تکنیکی به نام Direct Memory Access یا به طور خلاصه DMA استفاده می‌شود. این تکنیک به این صورت عمل می‌کند که وقتی پردازنده می‌خواهد یک بلوک داده را بخواند یا بنویسد، دستوری را به ماژول DMA ارسال می‌کند که شامل اطلاعات زیر است:

۱. اینکه آیا خواندن یا نوشتن درخواست شده است

۲. آدرس دستگاه I/O مرتبط

۳. مکان شروع در حافظه برای خواندن داده‌ها یا نوشتن داده‌ها

۴. تعداد کلماتی که باید خوانده یا نوشته شوند

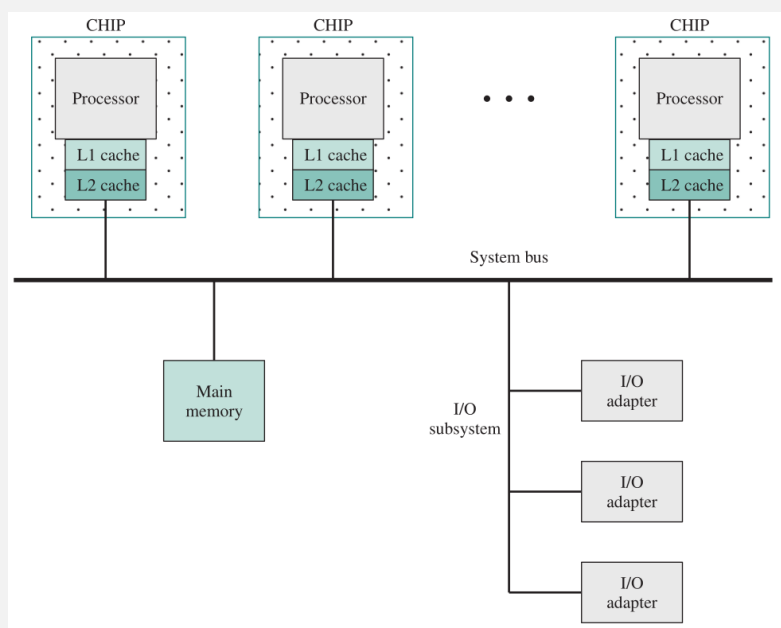
سپس پردازنده به کارهای دیگری ادامه می‌دهد. این عملیات I/O به ماژول DMA واگذار شده و آن ماژول آن را مدیریت خواهد کرد. ماژول DMA کل بلوک داده را به طور مستقیم به حافظه منتقل می‌کند بدون اینکه ابتدا آن را به پردازنده بدهد. وقتی انتقال کامل شد، ماژول DMA یک سیگنال وقفه به پردازنده ارسال می‌کند که پردازنده را متوجه پایان عملیات کند. بنابراین، پردازنده تنها در آغاز و پایان انتقال درگیر می‌شود.

سوال چهارم

نحوه عملکرد سیستم‌های چند پردازنده و سیستم‌های خوشه‌ای را توضیح دهید و آن‌ها را با یکدیگر مقایسه کنید. انواع دسته بندی آن‌ها را نیز نام ببرید.

پاسخ

۱. سیستم‌های چند پردازنده (Multi Processor): سیستم‌های چند پردازنده شامل چندین پردازنده در یک سیستم واحد هستند که به طور همزمان به پردازش داده‌ها می‌پردازند. این پردازنده‌ها از طریق یک حافظه مشترک و یک سیستم عامل واحد با هم در ارتباط هستند. هدف اصلی سیستم‌های چند پردازنده افزایش کارایی سیستم است، به طوری که پردازش‌ها بین پردازنده‌ها تقسیم می‌شود و به این ترتیب سرعت اجرای برنامه‌ها افزایش می‌یابد. در این نوع سیستم‌ها، همزمانی پردازش‌ها و مدیریت منابع به وسیله یک سیستم عامل واحد انجام می‌شود. از مزایای این سیستم می‌توان به بهبود کارایی، دسترسی پذیری بالا و تحمل خطا (Fault Tolerance) اشاره کرد، چرا که در صورت خرابی یک پردازنده، پردازنده‌های دیگر می‌توانند وظایف آن را بر عهده بگیرند. سیستم‌های چند پردازنده به دو دسته سیستم‌های چند پردازنده متقارن و غیر متقارن تقسیم می‌شوند. در سیستم‌های متقارن هر پردازنده تمام وظایف را انجام می‌دهد اما در سیستم‌های نامتقارن به هر پردازنده یک وظیفه مشخص اختصاص داده شده است. بلوک‌دیگرام یک سیستم چند پردازنده متقارن در زیر آورده شده است.

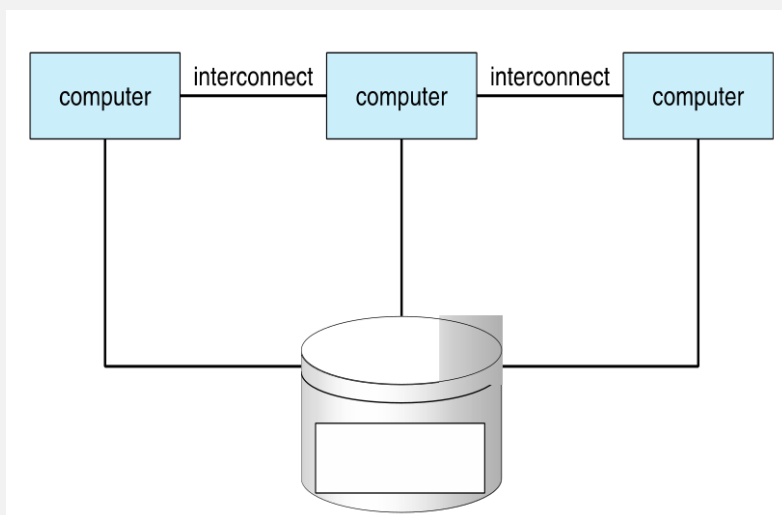


شکل ۴: سیستم‌های چندپردازنده متقارن

۲. سیستم‌های خوشه‌ای (Clustered Systems): سیستم‌های خوشه‌ای شامل مجموعه‌ای از چندین کامپیوتر مستقل هستند که به هم متصل شده‌اند و به عنوان یک سیستم واحد عمل می‌کنند. هر سیستم در این خوشه دارای پردازنده‌ها و حافظه مستقل است و با استفاده از شبکه به سیستم‌های دیگر متصل می‌شود. این نوع سیستم‌ها اغلب برای افزایش مقیاس پذیری (Scalability)، دسترسی پذیری بالا (High Availability) و قدرت محاسباتی استفاده می‌شوند.

پاسخ

یکی از ویژگی‌های بارز سیستم‌های خوشه‌ای این است که اگر یکی از سیستم‌ها از کار بیفتد، سایر سیستم‌ها می‌توانند همچنان به فعالیت خود ادامه دهند، که این امر باعث افزایش تحمل خطا و دسترس‌پذیری می‌شود. سیستم‌های خوشه‌ای به طور گسترده در مراکز داده و محاسبات علمی استفاده می‌شوند. نمونه‌ای از بلوک دیاگرام سیستم‌های خوشه‌ای در شکل زیر آورده شده است:



شکل ۵: سیستم‌های خوشه‌ای

این نوع سیستم‌ها نیز به دو دسته متقارن و نامتقارن تقسیم می‌شوند. در سیستم‌های متقارن همه سیستم‌ها همزمان باهم کار می‌کنند و همدیگر را چک می‌کنند اما در سیستم‌های نامتقارن یک سیستم آماده به‌کار دیگر وجود دارد که در صورت خرابی یک سیستم، جاگزین آن بشود.

سوال پنجم

موارد زیر و نحوه انجام آن‌ها در سیستم عامل را توضیح دهید.

۱. مدیریت زمان

پاسخ

مدیریت زمان در سیستم عامل به تخصیص منابع پردازشی به پردازنده‌ها و وظایف مختلف به صورت بهینه مربوط می‌شود. سیستم عامل باید مطمئن شود که همه پردازنده‌ها به طور منصفانه به پردازنده دسترسی داشته باشند و منابع به درستی بین پردازنده‌های مختلف تقسیم شوند. مدیریت زمان شامل مفاهیم زیر است:

(آ) CPU Scheduling: سیستم عامل با استفاده از الگوریتم‌های مختلف، پردازنده را به پردازنده‌ها تخصیص می‌دهد. الگوریتم‌هایی مانند FIFO و Round Robin و ... برای برنامه‌ریزی پردازنده‌ها استفاده می‌شوند.

(ب) Preemptive Scheduling: در این نوع برنامه‌ریزی، سیستم عامل می‌تواند پردازنده در حال اجرا را متوقف کند و به پردازنده دیگری اجازه اجرا دهد، تا اطمینان حاصل شود که پردازنده بین پردازنده‌ها به صورت عادلانه تقسیم می‌شود.

(ج) مدیریت تایمرها: سیستم عامل از تایمرها برای اجرای کارهایی در بازه‌های زمانی مشخص، مثل مدیریت دوره‌ای وظایف، استفاده می‌کند.

۲. مدیریت پردازنده‌ها

پاسخ

مدیریت پردازنده‌ها یکی از وظایف اصلی سیستم عامل است. سیستم عامل باید پردازنده‌ها را ایجاد، مدیریت و خاتمه دهد و منابع لازم را به آن‌ها تخصیص دهد. مراحل اصلی در مدیریت پردازنده‌ها عبارت‌اند از:

(آ) ایجاد پردازنده: زمانی که یک برنامه جدید اجرا می‌شود، سیستم عامل یک پردازنده برای آن ایجاد می‌کند و منابع لازم (مانند حافظه و دسترسی به ورودی/خروجی) را به آن تخصیص می‌دهد.

(ب) حالت‌های پردازنده: هر پردازنده می‌تواند در یکی از حالت‌های Ready، Running و Blocked باشد. سیستم عامل پردازنده‌ها را بین این حالت‌ها مدیریت می‌کند.

(ج) Context Switching: زمانی که سیستم عامل باید از یک پردازنده به پردازنده دیگری تغییر کند، باید وضعیت پردازنده فعلی (شامل ثبات‌ها و شمارنده برنامه) را ذخیره کند و سپس وضعیت پردازنده جدید را بازیابی کند. این عملیات به نام تعویض زمینه شناخته می‌شود.

(د) Process Termination: پس از پایان کار پردازنده، سیستم عامل آن را از حافظه خارج کرده و منابع تخصیص داده شده به آن را آزاد می‌کند.

۳. مدیریت حافظه

پاسخ

مدیریت حافظه در سیستم عامل به نحوه تخصیص و آزادسازی حافظه به پردازنده‌ها و برنامه‌های مختلف مربوط

پاسخ

می‌شود. سیستم عامل باید حافظه را به گونه‌ای مدیریت کند که پردازنده‌ها بتوانند به داده‌های خود دسترسی داشته باشند و از منابع به صورت بهینه استفاده شود. مراحل اصلی مدیریت حافظه شامل موارد زیر است:

(آ) تخصیص حافظه: سیستم عامل باید حافظه لازم برای اجرای پردازنده‌ها را به آن‌ها تخصیص دهد. این تخصیص می‌تواند به صورت تخصیص پیوسته ((Contiguous Allocation یا تخصیص غیرپیوسته (Non-Contiguous Allocation) انجام شود.

(ب) حافظه مجازی: سیستم عامل با استفاده از حافظه مجازی به پردازنده‌ها اجازه می‌دهد تا حافظه بیشتری از آنچه که به صورت فیزیکی در دسترس است، استفاده کنند. حافظه مجازی با تکنیک‌هایی مثل تبادل Paging یا Segmentation پیاده‌سازی می‌شود. این تکنیک‌ها به سیستم اجازه می‌دهند تا حافظه پردازنده‌ها را به قطعات کوچک‌تر تقسیم کند و آن‌ها را بین حافظه فیزیکی و دیسک جابه‌جا کند.

(ج) مدیریت فضای خالی: سیستم عامل باید از حافظه بهینه استفاده کند و فضاهای خالی حافظه را به طور مؤثر مدیریت کند. الگوریتم‌هایی مثل First Fit، Best Fit و Worst Fit برای تخصیص و مدیریت فضای خالی حافظه استفاده می‌شوند.

(د) حفاظت و دسترسی به حافظه: سیستم عامل با استفاده از تکنیک‌هایی مثل (Page Table و Pro-tecton Registers) از دسترسی غیرمجاز به بخش‌های مختلف حافظه جلوگیری می‌کند. این کار برای جلوگیری از تداخل پردازنده‌ها با یکدیگر و حفظ امنیت سیستم ضروری است.

سوال ششم

فرض کنید دو برنامه A و B در یک سیستم در حال اجرا هستند. به طور کلی هر نوع فعالیت مرتبط با حافظه ۳۰ میکروثانیه، اجرای ۶۰ دستورالعمل ۴ میکروثانیه و ۲۵ دستورالعمل ۲ میکروثانیه زمان می‌برد. بهره‌وری پردازنده هنگامی که سیستم قابلیت تک برنامه‌ای و چند برنامه‌ای دارد را محاسبه کنید و دیاگرام وضعیت پردازنده در واحد زمان را برای حالت چند برنامه‌ای رسم کنید.

A: Read a record from file
Executing 60 instructions
Write a record to file

B: Read a record from file
Executing 25 instructions
Write a record to file

پاسخ

۱. تک برنامه:

(آ) مجموع زمان اجرای برنامه A:

$$(2 \times 30^{\mu s}) + 4^{\mu s} = 64^{\mu s}$$

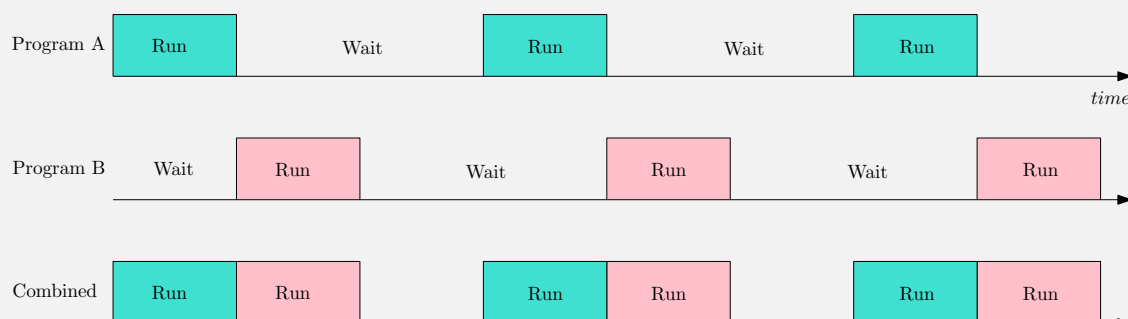
(ب) مجموع زمان اجرای برنامه B:

$$(2 \times 30^{\mu s}) + 2^{\mu s} = 62^{\mu s}$$

$$\rightarrow \text{CPU Utilization} = \frac{\text{Total Execution Time}}{\sum \text{Time}} = \frac{4 + 2}{64 + 62} = \frac{6}{126} = 0.0476 \approx 4.76\%$$

۲. چند برنامه:

در حالت چند برنامه، می‌توان در زمان‌های خالی CPU برنامه B را اجرا کرد. دیاگرام وضعیت پردازنده به صورت زیر می‌شود:



شکل ۶: دیاگرام زمان‌بندی CPU

پاسخ

بنابراین می‌توان از CPU به صورت بهینه استفاده نمود و در زمان‌های بیکاری سیستم (فاصله بین اتمام برنامه A تا شروع دوباره آن) می‌توان برنامه B را اجرا نمود. اما به دلیل آنکه در صورت سوال دوره تناوب اجرای هر برنامه مشخص نیست، مقدار ماکزیمم زمان اجرا را برای هر دو برنامه در نظر می‌گیریم و بهره‌وری نسبت به اجرای تک برنامه تغییری نمی‌کند! اما اگر تعداد اجرای هر برنامه در یک تناوب مشخص بود، اثبات می‌شد که با اجرای برنامه‌ها به پشت سر هم در زمان‌های بیکاری CPU بهره‌وری افزایش می‌یابد.

$$\text{CPU Utilization} = \frac{6}{126} = 0.0476 \approx 4.76\%$$

همچنین می‌توان اجرای دو برنامه را به صورت پایپلاین فرض نمود یعنی ابتدا هر برنامه عملیات I/O خود را انجام می‌دهد (خواندن و نوشتن)، سپس پردازنده دستورالعمل‌ها را اجرا می‌کند. پردازنده به تناوب بین برنامه A و B جابجا می‌شود. دیاگرام این روش به صورت زیر می‌شود:

0 – 30	30 – 60	30 – 34	60 – 90	90 – 92	92 – 122
Read A:	Read B:	Execute A:	Write A:	Write B:	Write B:

و در نهایت بهره‌وری CPU به صورت زیر محاسبه می‌شود:

$$\text{CPU Utilization} = \frac{6}{122} = 0.049 \approx 4.9\%$$