به نام خدا

با توجه به اهمیت موضوع سیستم‌های عامل در مهندسی کامپیوتر و IT، این مبحث در کلیه‌ی آزمون‌های ورودی دانشگاه‌ها در مقاطع مختلف و نیز آزمون‌های استخدامی و غیره جزو مباحثی است که مورد پرسش قرار می‌گیرد. لذا بدیهی است که یادگیری صحیح آن از اهمیت ویژه‌ای برخوردار است.
کتاب حاضر که با زحمات فراوان دوست ارجمندم جناب آقای مهندس موسوی تالیف گردیده است، برخلاف بسیاری از کتب موجود که متاسفانه هیچ کنترل علمی روی مطالب درسی و صحت پاسخ‌های تشریحی به عمل نیامده، از نظر اینجانب مورد تایید است و مطالب آن منطبق با آخرین مراجع معتبر دنیا مانند کتاب‌های تننباوم، استالینگز و سیلبرشاتز بوده و همچنان که کلیه‌ی مباحث و سرفصل‌ها را پوشش داده است, از انسجام خوبی نیز برخوردار است.
همچنین راه‌حل‌های ارائه شده برای مسائل و تست‌ها نشان می‌دهد که دقت بسیار زیادی در ارائه‌ی آن‌ها صورت گرفته که در صورت صحت اصل سوال، پاسخ‌ها صحیح بوده و راه‌حل‌ها می‌توانند با انتقال درست مفاهیم به دانشجویان، آنان را برای حل سوالات مشابه آماده نماید.

دکتر ابوالفضل طرقی حقیقت

ما چو نای ایم و نوا در ما ز توست
ما چو کوه ایم و صدا در ما ز توست
ما عدم هاییم و هستی های ما
تو وجود مطلقی، فانی نما
حضرت مولانا

نگارش کتابی با عنوان سیستم‌های عامل که افزون بر داشتن مفاهیم اساسی این درس دربردارنده‌ی نکات و تست‌های مطرح شده در کنکور کارشناسی ارشد نیز باشد، پیش از این مدنظر بود که اکنون با عنایت الهی این مهم میسر گشت. کتابی که پیش رو دارید، در شش فصل تدوین شده و بیش از ۶۰۰ سوال چهار گزینه‌ای از آزمون‌های کارشناسی ارشد و همچنین بالغ بر ۲۰۰ مساله‌ی برجسته و برگزیده از کتب مرجع، با حل تشریحی کامل و به‌صورت طبقه‌بندی گردآوری شده است. گرچه کتاب حاضر برای دانشجویانی که می‌خواهند بخت خویش را در آزمون‌های کارشناسی ارشد بیازمایند نگاشته شده اما با توجه به انطباق کامل مطالب آن با سرفصل وزارت علوم و شامل بودن مطالب کتب مرجع، می‌تواند برای دانشجویان درس سیستم‌عامل که دوست‌دار یادگیری و حل مسائل بیش‌تری در این زمینه هستند نیز سودمند باشد.

سعی برآن بوده که مطالب کتاب خالی از اشکال باشد، اما از آن‌جا که خطا قسمتی از زندگی ماست، مطالب این کتاب نیز مستثنا نخواهد بود. لذا از خوانندگان محترم خواهشمند است کاستی‌ها و اشتباهات کتاب را متذکر شوند و از ارائه‌ی نظرات و پیشنهادات سودمند خود از طریق پست الکترونیکی اینجانب دریغ نورزند.

از استاد ارجمند جناب آقای دکتر حقیقت بسیار سپاس‌گزارم که مطالب کتاب را مطالعه نمودند و نکات مهمی را درباره ساختار و محتوای کتاب متذکر شدند. همچنین از ایشان سپاس‌گزارم که با قلم دانشمندانه‌ی خود، سطوری را بر طلیعه‌ی این کتاب نقش زدند.

در انتها برخویش لازم می‌دانم از مهندسان کاس‌نژاد و مظاهری که ویراستاری کتاب را به نحو شایسته‌ای انجام دادند و از همسرم که با صبر و شکیبایی خود سهم ارزنده‌ای در تدوین این کتاب و دیگر مراحل زندگی بنده داشته‌اند و همچنین از مدیریت محترم انتشارات پوران پژوهش، جناب آقای دکتر هژبر و همکاران محترم ایشان، تشکر و قدردانی کنم.

امید است این کتاب بتواند مورد استفاده‌ی دانشجویان عزیز و اساتید محترم قرار گرفته و نقش بسزایی را در حل مشکلات دانشجویان و داوطلبان کنکور در درس سیستم عامل رشته‌ی کامپیوتر داشته باشد.

سید روح‌اله موسوی طیبی
پاییز ۹۹
moosavi_tayebi@yahoo.com

فهرست مطالب

فصل اول

مقدمه

یک سیستم کامپیوتری متشکل از سه بخش سخت‌افزار، نرم‌افزار و داده‌ها می‌باشد که بخش نرم‌افزار بـه دو دسـته نرم‌افزارهای سیستمی و نرم‌افزارهای کاربردی تقسیم می‌شود. سیستم‌عامل، به عنوان اساسی‌ترین نـرم‌افـزار سیسـتمی است که کامپیوتر را راه‌اندازی کرده و تا هنگامی که کامپیوتر روشن است، آماده انجـام وظـایف مـی‌باشـد. وظـایف سیستم‌عامل در دو دسته اصلی و مستقل **توسعه ماشین** و **مدیریت منابع** قرار مـی‌گیرنـد. در ادامـه هـر یـک از ایـن وظایف تشریح می‌شود.

سیستم‌عامل در نقش ماشین توسعه یافته[1] (ماشین مجازی[2])

در این نقش، سیستم‌عامل امکانات لازم را برای راحتی کاربران ارایه می‌دهد که در این راسـتا اهـداف زیـر محقـق می‌گردند:

۱) سیستم‌عامل، برنامه‌ای است که از یک طرف به‌عنوان واسط بین کاربر و برنامه‌های مد کاربر و از طرف دیگر بـه-عنوان واسط بین کاربر و منابع سیستم، انجام وظیفه می‌کند.

۲) سیستم کامپیوتری را برای استفاده آسان آماده می‌سازد.

۳) امکان اجرای برنامه‌های گرداننده[3]، به‌خصوص کنترل‌کننده‌های اجزای جانبی، و همچنین اداره وقفه‌هـای مربوطـه را فراهم می‌کند.

سیستم‌عامل در نقش مدیر منابع[4]

مهم‌ترین هدف از مطرح شدن سیستم‌عامل، مدیریت منابع است. در این نقش، سیستم‌عامل موجب اسـتفاده بهینـه از منابع و جلوگیری از هرج و مرج در به‌کارگیری اشتراکی آن‌ها می‌شود.

---

[1] Extended machine
[2] Virtual machine
[3] Driver
[4] Resource Manager

منابعی که سیستم‌عامل باید آن‌ها را مدیریت کند، در دو دسته اصلی زیر قرار می‌گیرند:

۱) منابع منطقی[1]: مانند فایل‌ها

۲) منابع فیزیکی[2]: مانند دستگاه‌های سخت‌افزاری پردازنده، حافظه اصلی و غیره

در مدیریت منابع، مالتی‌پلکس کردن (یا به کارگیری اشتراکی منابع) به دو روش انجام می‌شود: مالتی‌پلکس در زمان[3] و مالتی‌پلکس در فضا[4].

هنگامی که یک منبع مالتی‌پلکس زمانی می‌شود، برنامه‌های مختلف به ترتیب منبع را گرفته و پس از استفاده آن‌را رها کرده و به برنامه بعدی می‌دهند. به عنوان مثال، چند برنامه را در نظر بگیرید که باید توسط پردازنده اجرا شوند. برای این‌کار سیستم‌عامل ابتدا پردازنده را به یک برنامه اختصاص می‌دهد، هنگامی که برنامه به اندازه کافی از پردازنده استفاده کرد، برنامه دیگری شروع به استفاده از آن می‌کند و این روند تا آخرین برنامه ادامه پیدا می‌کند.

مالتی‌پلکس فضایی، نوع دیگر به کارگیری اشتراکی است که در آن به‌طور همزمان چندین برنامه در حال اجرا، هر کدام بخشی از منبع را در اختیار می‌گیرند. به عنوان مثال، حافظه اصلی را در نظر بگیرید که بین چندین برنامه در حال اجرا تقسیم می‌شود تا همه آن‌ها همزمان در آن قرار گیرند تا اگر نوبت دریافت پردازنده یک برنامه فرا رسید، به سرعت به آن برنامه سوییچ شود. به عبارت دیگر، اگر در حافظه اصلی فضای کافی برای نگه‌داری هم‌زمان چندین برنامه وجود داشته باشد، مالتی‌پلکس فضایی، کارایی حافظه را نسبت به روشی که فقط یک برنامه را در حافظه قرار می‌دهد، به طور موثری افزایش خواهد داد.

برای درک بیش‌تر مفهوم، مراحلی را که سیستم‌عامل، در هنگام درخواست یک منبع توسط یک یا چند برنامه‌ی در حال اجرا، باید طی کند، آورده شده است:

۱) بررسی وضعیت منبع[5]

اصولا در سیستم کامپیوتری، یک منبع دو وضعیت می‌توانند داشته باشند: ۱) تخصیص داده شده ۲) آماده تخصیص (آزاد). روال کار به این صورت است که در حافظه اصلی نقشه‌ای از وضعیت منابع سیستم وجود دارد و برای تخصیص، لازم است که سیستم‌عامل ابتدا به آن مراجعه کند. در این نقشه نام تمامی منابع، نمونه‌های آن و وضعیت‌شان مشخص شده است و سیستم‌عامل طبق آن می‌تواند تصمیم بگیرد که منبع قابلیت تخصیص دارد یا خیر.

۲) زمان‌بندی[6]

وقتی در یک سیستم‌عامل، چند برنامه‌ی در حال اجرا به طور همزمان درخواست یک منبع را داشته باشند، باید زمان‌بندی انجام گیرد (مالتی‌پلکس زمانی). به عبارت دیگر، زمان‌بندی یعنی کدام یک از منابع سیستم، در چه زمانی

---

[1] Logical resources
[2] Physical resources
[3] Time Multiplexing
[4] Space Multiplexing
[5] Resource Status
[6] Scheduling

و به کدام برنامه، تخصیص یابد. نکته قابل توجه این است که پردازنده تنها منبعی است کـه زمـان‌بنـدی آن بـه طـور مستقل و بر دو گونه انحصاری و غیرانحصاری انجام می‌شود.[۱]

۳) تخصیص منبع[۲]

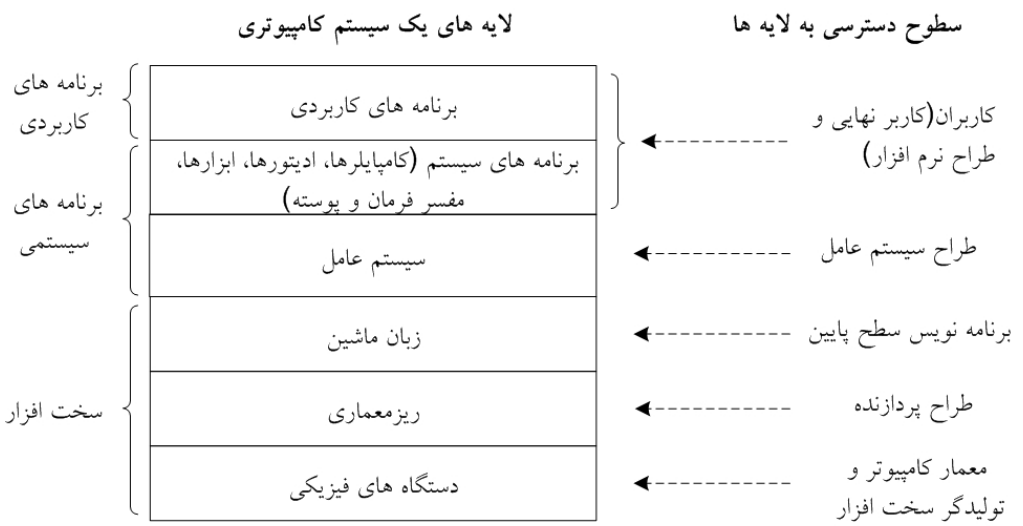
زمانی که منبعی در اختیار یک برنامه در حال اجرا قرار می‌گیرد، وضعیت آن از آزاد به تخصـیص داده شـده تغییـر می‌یابد تا از در اختیار گذاشتن آن به سایر برنامه‌ها جلوگیری شود.

۴) آزادسازی منبع[۳]

پس از آن‌که کار با یک منبع تمام شد، وضعیتش به آماده تخصیص (یا آزاد) تبدیل می‌شود تا سایر برنامه‌ها بتواننـد در صورت نیاز از آن استفاده کنند.

۱-۱ دید کلی نسبت به سیستم‌عامل

همان‌طور که در شکل ۱-۱ ملاحظه می‌شود، می‌توان یـک سیسـتم کـامپیوتری را بـه‌صـورت لایـه‌ای و سلسـله مراتبی دید:



شکل ۱-۱: لایه‌های یک سیستم‌کامپیوتری

در پایین‌ترین لایه، **دستگاه‌های فیزیکی** قرار دارند که شامل تراشه‌های مدار مجتمع، سیم‌ها، منـابع تغذیـه و مـوارد دیگر است که طرز کار و چگونگی ساخت آن‌ها خارج از حوصله این کتاب می‌باشـد. در بـالای آن، **ریزمعمـاری** قرار دارد. این سطح شامل ثبات‌های درون پردازنده و یک مسیر داده[۱] است.

---

۱ در فصل دوم، زمان‌بندی انحصاری و غیرانحصاری توضیح داده می‌شود.

[2] Resource Allocation
[3] Resource Reallocation

مسیر داده، هسته اصلی پردازنده است که تمامی محاسبات در آن انجام می‌گیرد. در بعضی از کامپیوترها، مسیر داده توسط نرم‌افزاری به نام ریزبرنامه، کنترل می‌شود. ریزبرنامه معمولا در حافظه فقط خواندنی[۲] قرار دارد و در واقع یک مفسر است که دستورالعمل‌های زبان ماشین مانند Add، Move و Jump را واکشی کرده و هر یک از آن‌ها را در یک سری از قدم‌های کوتاه‌تر انجام می‌دهد. به این کامپیوترها، کامپیوترهای با دستورات زیاد CISC[۳] می‌گویند که در مقابل آن‌ها کامپیوترهای کم دستور RISC[۴] قرار دارند. در کامپیوترهای RISC سطح ریزبرنامه وجود ندارد و سخت‌افزار مستقیما دستورالعمل‌های زبان ماشین را اجرا می‌نماید. از دید زبان اسمبلی، سخت‌افزار و دستورالعمل‌های آن، معماری مجموعه دستورالعمل‌ها[۵] را تشکیل می‌دهند و اغلب به این سطح، **زبان ماشین** گفته می‌شود. زبان ماشین معمولا بین ۵۰ تا ۳۰۰ دستورالعمل دارد که اکثر آن‌ها برای جابه‌جایی داده‌ها در ماشین، عملیات محاسباتی و مقایسه مقادیر به کار می‌روند. در این لایه، دستگاه‌های I/O به وسیله بار کردن مقادیر در ثبات‌های ویژه دستگاه[۶]، کنترل می‌شوند.

در سطح بعد **سیستم‌عامل** قرار دارد و همان‌طور که اشاره شد، وظیفه آن مخفی نمودن همه این پیچیدگی‌ها و ارایه یک مجموعه دستورالعمل راحت‌تر به برنامه‌نویس است. در بالای سیستم‌عامل، سایر **نرم‌افزارهای سیستمی** قرار دارند که شامل مفسرهای فرمان (پوسته)، ابزارها[۷]، کامپایلرها، مفسرها، ادیتورها و سایر برنامه‌های مستقل از کاربرد هستند. این برنامه‌ها خارج از سیستم‌عامل هستند، گرچه گاهی توسط سیستم‌عامل یا سازندگان کامپیوتر عرضه می‌شوند، اما باید توجه داشت که سیستم‌عامل آن بخش از نرم‌افزار است که در مد هسته[۸] (مد راهبری[۹]) اجرا می‌شود و در مقابل، کامپایلرها و ادیتورها در مد کاربر[۱۰] اجرا می‌شوند. (پردازنده دو مد (حالت) اجرا دارد: مد هسته و مد کاربر. در مد هسته، دستورالعمل‌های ممتاز اجرا می‌شوند مانند دستورالعمل‌های مدیریت حافظه و دستورالعمل‌های I/O و غیره، ولی در مد کاربر، اجرای برنامه‌های کاربر صورت می‌گیرد).

در واقع **هسته‌ی سیستم‌عامل** روی سخت‌افزار می‌نشیند تا لایه‌ها و سطوح بعدی را از سخت‌افزار جدا نگه دارد. در هسته‌ی سیستم‌عامل تمامی قسمت‌های بنیادی و پایه‌ای مانند مدیریت حافظه، ورودی و خروجی و غیره تعریف می‌شوند. این بخش به کمک سخت‌افزار در مقابل مداخله کاربران محافظت می‌شود. برای این منظور لازم است سیستم‌عامل قبل از تخصیص پردازنده به برنامه کاربر، پردازنده را به مد کاربر سوییچ کرده تا چنان‌چه برنامه در طی اجرا (و البته زمانی که سیستم‌عامل در حال اجرا نبوده و هیچ قدرتی ندارد) پا را از گلیم خود فراتر گذاشت و کارهای غیر مجازی مانند اجرای دستورالعمل‌های ممتاز (مانند از کار انداختن وقفه‌ها)، دسترسی بدون اجازه به محدوده حافظه سایر برنامه‌های دیگر (از جمله برنامه سیستم‌عامل)، دسترسی مستقیم به ثبات‌های ویژه دستگاه‌های ورودی و خروجی و غیره را انجام داد، پردازنده آن‌ها را کشف و از ادامه آن برنامه خودداری کند و با ورود به مد

---

[1] Data Path
[2] Rom
[3] Complex Instruction Set Computer
[4] Reduced Instruction Set Computer
[5] ISA (Instruction Set Architecture)
[6] Special device registers
[7] Tools
[8] Kernel mode شرح مد کاربر و مد هسته، در انتهای همین فصل، مبحث عملکرد دوگانه، آمده است.
[9] Supervisor mode
[10] User mode

هسته به اجرای اداره کننده وقفه مربوط به استثنای پیش آمده بپردازد تا سیستم‌عامل تکلیف برنامه مذکور را روشن کرده و مثلا آن‌را از بین ببرد.

نکته‌ای که لازم است در این‌جا به آن اشاره شود، این است که سیستم‌عامل از درون برنامه‌هایی که انجام می‌شوند، هیچ اطلاعی ندارد و فقط می‌داند که مثلا برنامه x، منبع y را نیاز دارد (وظیفه پردازنده نیز، تنها اجرای دستورالعمل‌هاست و هیچ اطلاعی از درون آن‌ها ندارد). از جمله موجودیت‌هایی که می‌توانند از درون برنامه‌ها اطلاعی داشته باشند، کامپایلر و اسمبلر هستند.

نهایتا در بالای برنامه‌های سیستمی، **برنامه‌های کاربردی** قرار دارند که شامل سیستم‌های بانک اطلاعاتی، مرورگرهای وب[1]، واژه‌پردازها[2]، بازی‌ها و غیره می‌باشند.

جایگاه سیستم‌عامل در حافظه اصلی

از لحاظ ساختمان داده‌ها، فضای حافظه اصلی به صورت آرایه‌ای خطی از بایت‌ها می‌باشد که هر بایت از آن، یک آدرس غیرتکراری دارد. برای اجرای سیستم‌عامل و برنامه‌ها لازم است در حافظه اصلی دو ناحیه ایجاد گردد: ۱) ناحیه‌ای شامل سیستم‌عامل، که به آن King Space گفته می‌شود. ۲) ناحیه‌ای که فقط برنامه‌های کاربر را پوشش می‌دهد و اصطلاحا به آن User Space می‌گویند.

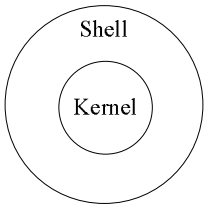
تعریف فرایند

فرایند، برنامه‌ای است که توسط زمان‌بند کار[3] انتخاب و وارد چرخه اجرا شده، ولی هنوز پایان نیافته و از سیستم خارج نشده است. توجه داشته باشید که الزامی در این‌که یک فرایند هر لحظه پردازنده و یا حافظه اصلی را در اختیار داشته باشد، وجود ندارد. همان‌طور که در فصل بعد خواهید دید، یک فرایند ممکن است به دلیل عمل تعلیق، به‌طور موقت مجبور به قرار گرفتن در حافظه جانبی باشد.

در سایر متون فارسی به‌جای فرایند از واژه‌های دیگری مانند فرارونده، پردازه، پردازش، پروسه و پروسس نیز استفاده می‌شود.

بخش‌های اصلی سیستم‌عامل

مطابق شکل زیر یک سیستم‌عامل از دو بخش اصلی هسته (Kernel) و پوسته ارتباطی (Shell) تشکیل می‌شود.



شکل ۱-۲: دو بخش اصلی سیستم‌عامل

[1] Web browsing
[2] Work Processing
[3] Job Scheduler

پوسته ارتباطی، واسط بین کاربر با هسته سیستم‌عامل و همچنین برنامه‌های کاربر بوده که وظیفه آن ترجمه و تفسیر دستورات به زبان قابل فهم برای هسته است. پوسته ارتباطی به عنوان **مفسر فرمان** شناخته می‌شود. مفسر فرمان در سیستم‌عامل‌های DOS و Unix، محل تایپ فرمان و در سیستم‌عامل Windows، گرافیکی (GUI)[1] می‌باشد که به‌جای تایپ در خط فرمان، با انجام کلیک بر روی آیکون برنامه، اجرای آن آغاز می‌شود.

هسته، همان سیستم‌عامل است که بخش‌های مدیریتی در آن قرار می‌گیرند. محل قرارگیری هسته روی سخت‌افزار است تا به این وسیله، لایه‌های دیگر را از سخت‌افزار جدا نگه دارد.

فراخوان‌های سیستمی (System Call): واسط بین برنامه‌های کاربردی در حال اجرا و هسته سیستم‌عامل بوده و اغلب توابعی به زبان اسمبلی هستند که می‌توانند مستقیما با سخت‌افزار ارتباط برقرار کنند.

اگر یک فرایند مشغول اجرای یک برنامه در مد کاربر باشد و نیاز به یک سرویس سیستمی (مانند خواندن داده از یک فایل) داشته باشد، مجبور خواهد بود که یک تله[2] یا فراخوان سیستمی را اجرا کند تا کنترل ماشین را از مد کاربر به مد هسته برده و به سیستم‌عامل بسپارد. سپس سیستم‌عامل با توجه به پارامترهای ارسال شده، متوجه خواسته فرایند صدا زننده می‌شود، آنگاه فراخوان سیستمی را اجرا کرده و پس از انجام آن، کنترل را به دستورالعمل بعد از فراخوان سیستمی بر می‌گرداند. در نتیجه، فراخوان سیستمی شبیه یک نوع خاص از فراخوان رویه‌های معمولی به نظر می‌رسد، با این تفاوت که فراخوان سیستمی وارد مد هسته یا راهبری می‌شود اما فراخوان رویه‌های معمولی این‌گونه نیست. فراخوان‌های سیستمی در سیستم‌های‌عامل ویندوز مایکروسافت تحت عنوان API[3] و در برخی از سیستم‌های قدیمی تحت عنوان SVC[4] شناخته می‌شوند.

بخش‌های مدیریتی سیستم‌عامل

۱) مدیریت فرایند[5]

یکی از اصلی‌ترین بخش‌های مدیریتی سیستم‌عامل، مدیریت فرایند است. مدیریت فرایند از ساخت یک فرایند تا خاتمه آن‌را مدیریت کرده و شامل موارد زیر می‌باشد:

ایجاد و حذف فرایندها، زمان‌بندی فرایندها برای دریافت منابع از جمله پردازنده، مدیریت همزمانی و همگام‌سازی فرایندها[6]، ارتباط بین فرایندها[7] و جلوگیری از بن‌بست[8].

بخش مدیریت فرایند، به تفصیل در فصول ۲ و ۳ و ۴ بررسی می‌شود.

۲) مدیریت حافظه اصلی و مدیریت حافظه انبوه (دیسک)[9]

در بخش مدیریت حافظه اصلی، سیستم‌عامل اعمال زیر را انجام می‌دهد:

---

[1] Graphical user interface
[2] Trap
[3] Application program interface
[4] Supervisor call
[5] Process management
[6] Process concurrency and synchronization
[7] Process communication
[8] Deadlock handling
[9] Memory – Disk management

- تخصیص فضای حافظه به فرایندها و بازپس‌گیری آن‌ها.
- نگه‌داری بخش‌های آزاد حافظه اصلی و همچنین مشخص کردن بخش‌های تخصیص یافته به فرایندها به همراه نام آن‌ها.
- در زمان آزاد شدن فضای حافظه اصلی، تصمیم‌گیری در این مورد که چه فرایندی به حافظه بارگذاری شود.

در ارتباط با مدیریت دیسک، از آن‌جا که ممکن است اطلاعات یک فایل بر روی سطح دیسک پخش باشد، چگونگی و ترتیب دسترسی به سکتورها (Disk Scheduling) و همچنین مدیریت فضای آزاد دیسک برعهده سیستم‌عامل است.

لازم به ذکر است که در ارتباط با عمل مبادله (Swapping) [۱] نیز سیستم‌عامل وظیفه تخصیص و رهاسازی بخش‌های حافظه و حفاظت از تداخل فرایندها را عهده‌دار است.

۳) مدیریت فایل [۲]

مدیریت فایل در سیستم‌عامل، شامل موارد زیر است:

ایجاد و حذف فایل‌ها، دایرکتوری‌ها (فهرست راهنما)، انجام عملیات کپی، انتقال و تغییرات بر روی آن‌ها، ذخیره‌سازی، پشتیبان‌گیری و مدیریت قرارگیری فایل‌ها بر روی رسانه‌های ذخیره‌سازی، مدیریت سطوح دسترسی به فایل‌های مشترک و انجام نگاشت [۳] فایل‌ها بر روی ذخیره ثانوی.

۴) مدیریت سیستم‌های ورودی و خروجی [۴]

سیستم‌عامل محیطی را فراهم می‌کند تا کاربر از پیچیدگی سخت‌افزاری کار با وسایل ورودی و خروجی مبرا باشد. عملیاتی مانند مدیریت بافرها، اسپولینگ، اجرای درایورهای [۵] وسایل مختلف، جلوگیری از تداخل کاری وسایل ورودی و خروجی و اداره بن‌بست‌ها برعهده سیستم‌عامل است.

۵) شبکه سازی [۶]

انواع ارتباطات شبکه‌ای نیاز به مدیریت سیستم‌عامل داشته و شبکه‌سازی بخشی از سیستم‌عامل را شامل می‌شود. در این حیطه، سیستم‌عامل موارد زیر را کنترل می‌کند: دسترسی به شبکه، دسترسی به فایل‌ها در یک محیط مشترک (Sharing)، مسیریابی، کنترل تراکم، تامین ارتباط بین پردازنده‌ها و کامپیوترها.

سیستم‌عامل علاوه بر ۵ بخش اصلی و مدیریتی فوق، وظایفی نیز بر عهده دارد که در ادامه هر یک از آن‌ها بررسی می‌شود:

---

[۱] عمل مبادله در فصل دوم بررسی می‌شود.

[2] File Management
[3] Mapping
[4] I/O System Management

[۵] تمامی دستگاه‌های متصل به یک کامپیوتر نیازمند یک کد خاص برای کنترل آن دستگاه می‌باشند. این کد گرداننده (درایور) دستگاه نامیده می‌شود.

[6] Networking

۱) سیستم مفسر فرمان[1]

در قسمت Shell توضیح داده شد که سیستم مفسر فرمان هم به‌صورت text و هم به‌صورت گرافیکی، برای گرفتن دستور از کاربر و برگرداندن نتیجه کار به کاربر استفاده می‌شود، که این کار از جمله وظایف سیستم‌عامل است.

۲) ایجاد یک محیط بدون خطا

سیستم‌عامل سعی می‌کند که محیط کاری ایجاد شده برای کاربر، یک محیط بدون خطا باشد و این امر تنها در سیستم‌عامل‌های بلادرنگ، به طور کامل وجود دارد.

۳) کنترل اشتباهات[2]

اگر برنامه‌ای از لحاظ کامپایلری دچار مشکل نباشد، اما در اجرا خطاهایی مانند سرریز پشته یا تقسیم بر صفر داشته باشد[3]، وظیفه سیستم‌عامل است که با یک وقفه، پردازنده را از اجرای آن برنامه باز دارد.

۴) امنیت[4]

به این معناست که سیستم در برابر حملات مختلف (ویروس، کرم رایانه‌ای و غیره) مقاوم باشد. اکثر سیستم-عامل‌ها، سرویس‌های لازم را در این زمینه دارند و البته بعضی از نرم‌افزارها برای ایجاد امنیت بیش‌تر در این زمینه استفاده می‌شوند.

۱-۲ بررسی سیر تکاملی سیستم‌عامل‌ها

سیر تکاملی سیستم‌عامل‌ها بر مبنای نیازهای جدید، تکامل سخت‌افزار و به‌کارگیری معماری جدید در ساخت یک سیستم کامپیوتری پدید آمده است که در ادامه از سه منظر کاربردی، صنعتی و ساختاری بررسی می‌شود.

۱-۲-۱ بررسی سیستم‌عامل‌ها از لحاظ کاربردی

۱) لامپ‌های خلا و مدارهای سوراخ‌دار (نسل اول ۵۵-۱۹۴۵)

ماشین‌های اولیه از لامپ خلا بهره می‌بردند و در آن‌ها از هیچ سیستم‌عاملی استفاده نمی‌شد. فقط یک گروه از متخصصین تمامی مراحل طراحی، ساخت، برنامه‌نویسی، استفاده و نگه‌داری از ماشین را بر عهده داشتند. در آن زمان کلیه برنامه‌ها به زبان ماشین نوشته می‌شد و در بیشتر ماشین‌ها برنامه‌نویسان، برنامه زبان ماشین (۰ و ۱) خود را به وسیله سیم‌بندی تخته مدارهای سوراخ‌دار، که ورودی ماشین محسوب می‌شد، پیاده‌سازی می‌کردند. این برنامه‌ها، کنترل روند اجرای عملیات پایه‌ای ماشین را بر عهده داشتند.

---

[1] Command Interpreter
[2] Fault Monitoring
[3] Run time errors
[4] Security

پس از مدتی این روال با ظهور کارت‌های منگنه[1] بهبود یافت که در آن برنامه‌ها به‌جای تخته مدارهای سوراخ‌دار، بر روی این کارت‌ها منگنه می‌شد و کامپیوترها توسط دستگاه کارت‌خوان، برنامه‌ها را می‌خواندند.

معایب این گونه سیستم‌ها عبارتند از:

۱) این ماشین‌ها سیستم‌عامل نداشتند.

۲) از سرعت پایینی برخوردار بودند.

۳) به‌دلیل احتمال بالای سوختن لامپ‌های خلا در حین انجام عملیات، کار با آن‌ها ریسک بالایی داشت.

۴) این سیستم‌ها حجیم و بزرگ بودند.

۵) با کاربر تعامل نداشتند.

۲) سیستم‌های ساده دسته‌ای[2] (نسل دوم ۶۵-۱۹۵۵)

با ظهور ترانزیستور که منجر به پیدایش این سیستم‌ها شد، قابلیت اطمینان کامپیوترها به نحو چشم‌گیری افزایش یافت. ایده کار در سیستم‌های دسته‌ای به این صورت بود که ابتدا برنامه‌های مختلف بر روی کارت‌ها منگنه می‌شد و سپس آن‌هایی که دارای نیاز یکسان بودند (مثل نیاز به کامپایلر یکسان) در یک گروه، به یک کارت خوان داده می‌شد تا آن‌ها را توسط دستگاه نوارگردان بر روی یک نوار مغناطیسی ذخیره کند.

اجرای برنامه‌ها ایجاب می‌کرد ابتدا اپراتور، یک برنامه مخصوص (جد سیستم‌های عامل امروزی) را بار کند تا اولین کار را از روی ورودی خوانده و اجرا نماید. خروجی کار بر روی یک نوار دیگر نوشته می‌شد و پس از اتمام هر کار، سیستم‌عامل به‌صورت خودکار کار بعدی را از نوار خوانده و شروع به اجرای آن می‌کرد.

با توجه به ارتباط غیر مستقیم برنامه‌نویس با ماشین، در این سیستم‌ها کار تعریف خاصی داشت: **کار (Job)** عبارت بود از مجموعه‌ای شامل برنامه(ها)، داده‌های ورودی و نیز فرامین کنترل کار. وجود این فرمان‌ها که به زبان‌های خاص کنترل کار JCL[3] نوشته می‌شدند، برای کنترل روند بارگذاری کامپایلر و برنامه کاربر، کامپایل و نیز اجرای برنامه، ضروری بود.

لازم به‌ذکر است که در برخی از متون سیستم‌عامل، تکنیک دسته‌ای فوق را Offline Spooling می‌نامند.

مزایای این گونه سیستم‌ها عبارتند از:

۱) به دلیل استفاده از نوارگردان‌های ورودی و خروجی به‌جای کارت‌خوان و چاپگر، زمان بی‌کاری پردازنده در هنگام خواندن کار و چاپ نتایج، نسبت به سیستم‌های قبلی کاهش یافت.

۲) نسبت به کامپیوترهای نسل اول، راندمان بهتر و عملیات ساده‌تر داشتند.

۳) در این سیستم‌ها، امکان انتقال اتوماتیک یک کار به کار دیگر وجود داشت.

---

[1] Punched card
[2] Simple Batch systems
[3] Job Control Language

معایب این گونه سیستم‌ها عبارتند از:

۱) زمان گردش کار (زمان برگشت)[1]، طولانی بود (به لحظه تحویل کار به سیستم تا لحظه آماده شدن خروجی جهت تحویل به کاربر، زمان گردش کار گفته می‌شود).

۲) در این سیستم‌ها تعامل با کاربر وجود نداشت. از جمله مشکلات برنامه‌نویسان در این سیستم‌ها، رفع یکایک خطاهای برنامه بود که امکان داشت چندین مرحله طول بکشد و عدم امکان اجرای گام به گام برنامه و مشاهده مقادیر متغیرها در طی اجرا، باعث بروز این مشکل می‌شد. و چنان‌چه نقص و خطایی در برنامه‌ی در حال اجرا رخ می‌داد، به‌جای خروجی برنامه، محتویات ثبات و حافظه چاپ می‌گشت.

۳) به خاطر اختلاف سرعت زیاد پردازنده با دستگاه نوارگردان، درصد بیکاری پردازنده همچنان بالا بود.

۴) برای اولویت دادن به کارهای مهم، روشی جز چیدن دستی آن‌ها قبل از انتقال به نوار ورودی وجود نداشت. همچنین امکان فرستادن سریع یک کار مهم تازه وارد برای اجرا، فراهم نبود.

۵) این سیستم‌ها نیز حجیم و بزرگ بودند.

۶) در این سیستم‌ها امکان اجرای یک کار و I/O به‌طور همزمان وجود نداشت.

۳) سیستم‌های چند برنامه‌ای[2] (نسل سوم ۸۰-۱۹۶۵)

از جمله مشکلاتی که در سیستم‌های قبل وجود داشت، این بود که وقتی یک کار برای تکمیل عملیات I/O (مثل نوارگردان) منتظر می‌ماند، پردازنده بی‌کار می‌شد تا عملیات I/O به اتمام برسد. راه حلی که در سیستم‌های چندبرنامه‌ای ارائه شد، به این صورت است که ابتدا فضای حافظه به چند تکه تقسیم شده و هر کار در یک پارتیشن قرار می‌گیرد. وقتی که یک کار برای تکمیل عملیات I/O منتظر می‌ماند، بلافاصله یکی دیگر از کارهای درون حافظه، پردازنده را در اختیار می‌گیرد. چنان‌چه آن کار هم به I/O نیاز پیدا کند، پردازنده به کار دیگری سوییچ می‌کند و به همین ترتیب تا کار آخر.

هنگامی که عملیات I/O کار اول به اتمام رسید، پردازنده مجددا به آن کار تخصیص داده می‌شود. اگر تعداد کارهای موجود در حافظه کافی باشد، می‌توان پردازنده را تقریبا صددرصد مشغول نگه داشت.

لازم به‌ذکر است که در ساخت این سیستم‌ها، از مدارات مجتمع (IC) استفاده می‌شود و همچنین نگه‌داری همزمان چند کار درون حافظه، نیاز به سخت‌افزار مخصوص جهت حفاظت از هر کار در برابر دسترسی پنهانی سایر کارها دارد.

تصویر حافظه اصلی در این نوع سیستم‌ها مطابق شکل ۱-۳ است.

---

[1] Turnaround time
[2] Multi programming systems

شکل ۱-۳: یک سیستم چندبرنامه‌ای با سه کار درون حافظه

شایان ذکر است که در این سیستم‌ها، کارت‌ها مستقیما به جای نوار مغناطیسی، به دیسک منتقل می‌شوند و هرگاه یک کار در حال اجرا به پایان برسد، سیستم‌عامل می‌تواند یک کار جدید را از روی دیسک برداشته و در یک بخش خالی شده از حافظه بار کرده و سپس آن را به اجرا در آورد. این تکنیک Spooling (یا Online spooling) نامیده می‌شود. در این تکنیک نیازی به نوار مغناطیسی و نوار گردان نیست.

مزایای این گونه سیستم‌ها عبارتند از:

۱) در این سیستم‌ها، پردازنده همواره مشغول پردازش است.

۲) اولین نمونه‌ای در سیستم‌عامل است که برای کاربران از لحاظ انجام کار تصمیم می‌گیرد.

۳) ایده زمان‌بندی کار و زمان‌بندی پردازنده در این سیستم‌ها مطرح می‌شود.

۴) در این سیستم‌ها، از تکنیک دسترسی مستقیم به حافظه[1] استفاده می‌شود که در آن پردازنده می‌تواند برای انجام یک کار ورودی و خروجی، یک فرمان به DMA صادر کند و پس از سپردن کار به آن، خود به پردازش کار دیگری بپردازد. به‌عبارتی امکان اجرای همزمان کار با I/O وجود دارد.

معایب این گونه سیستم‌ها عبارتند از:

۱) همانند آن‌چه در سیستم‌های دسته‌ای گفته شد، هنوز تعامل بین کاربر (برنامه‌نویس) و سیستم وجود ندارد.

۲) سیستم‌عامل‌های چندبرنامه‌ای به‌دلیل نیاز به مدیریت حافظه، از سیستم‌عامل‌های ساده تک برنامه‌ای پیچیده‌تر هستند.

۳) این سیستم‌ها نیز حجیم و بزرگ هستند.

[1] DMA (Direct Memory Access)

۴) سیستم‌های اشتراک زمانی[1] (چند وظیفه ای[2])

همان‌طور که بررسی شد، سیستم‌های دسته‌ای با استفاده از عملکرد چندبرنامه‌ای، مفید واقع می‌شوند. اما در بسیاری از کارها تعامل[3] کاربر با سیستم ضروری است. مبنای سیستم‌های اشتراک زمانی این است که تعدادی کاربر وجود دارند که می‌خواهند از طریق پایانه‌های خود به‌طور همزمان از سیستم (mainframe) استفاده کنند و در بین اجرای هر دو برنامه (کاربر)، سیستم‌عامل برای مدت کوتاهی اجرا می‌گردد.

به طور مثال اگر n کاربر در سیستم وجود داشته باشند، زمان پردازنده بین آن‌ها تقسیم و به اشتراک گذاشته می‌شود به گونه‌ای که به هر کدام تقریبا 1/n از زمان کل پردازنده می‌رسد. این زمان اصطلاحا برش زمانی[4] یا کوانتوم[5] نامیده می‌شود. عمل تخصیص برش‌های زمانی به کاربران، آن‌قدر سریع انجام می‌گیرد که امکان کار محاوره‌ای آن‌ها با سیستم به وجود می‌آید. لازم بذکر است که برای تضمین اجرای هر کار در مدت برش زمانی تعیین شده، نیاز به استفاده و تنظیم وقفه ساعت[6] (Timer) می‌باشد.

در واقع سیستم‌های اشتراک زمانی نوع خاصی از سیستم‌های چند برنامه‌ای هستند که در آن‌ها تعویض کار بر اساس یک معیار زمانی (بازه زمانی) و نه بر اساس نیاز آن کار به ورودی و خروجی، صورت می‌گیرد.

مزایای این گونه سیستم‌ها عبارتند از:

۱) امکان تعامل کاربر با برنامه وجود دارد.

۲) استفاده چند کاربر به طور همزمان از یک کامپیوتر امکان‌پذیر است.

۳) در سیستم‌های اشتراک زمانی، فرمان‌هایی که از طریق پایانه وارد می‌شوند، منبع دستورات به سیستم‌عامل هستند، نه دستورالعمل‌های زبان کنترل کار.

معایب این گونه سیستم‌ها عبارتند از:

۱) این سیستم‌ها حجیم و بزرگ هستند.

۲) نیاز به دقت در تنظیم برش‌های زمانی وجود دارد.

۵) کامپیوترهای شخصی[7] (نسل چهارم ۱۹۸۰ تا کنون)

با توسعه مدارات مجتمع با مقیاس بزرگ (LSI)[8] که تراشه‌هایی شامل هزاران ترانزیستور در یک سانتی‌متر مربع از سیلیکن بود، کامپیوترهای شخصی مبتنی بر ریزپردازنده[9] به وجود آمد.

---

[1] Time Sharing
[2] Multi tasking
[3] Interactive
[4] Time Slice
[5] Quantum

[6] این نوع وقفه، جلوتر در دسته‌بندی وقفه شرح داده می‌شود.

[7] Personal Computer
[8] Large-Scale Integrated circuit
[9] Microprocessor

در این کامپیوترها از یک طرف کارت‌خوان‌ها با صفحه کلید و ماوس، و از طرفی دیگر چاپگرهای بزرگ با صفحه نمایش و چاپگرهای کوچک، جایگزین شدند. این تغییر و تحولات در سخت‌افزار، سبب کاهش قیمت کامپیوترها و به نوعی راحتی کار با آن‌ها شد. سیستم‌عامل‌های اولیه بر روی PCها فقط تک کاربره و تک برنامه‌ای بودند مانند سیستم‌عامل DOS، ولی سیستم‌عامل‌های امروزی مانند Windows، خاصیت‌های چندبرنامگی، چند کاربره و شبکه‌ای را نیز دارند. این سیستم‌عامل‌ها به تدریج در طول زمان تغییر نمودند و به‌جای ماکزیمم نمودن درصد استفاده از پردازنده و وسایل جانبی، به سمت راحتی کاربر پیش رفتند.

با پیشرفت این سیستم‌ها، ویژگی‌های مهم سیستم‌عامل‌های قدیمی در Mainframe، از قبیل حفاظت حافظه، حافظه مجازی، همزمانی فرایندها و غیره، بر روی سیستم‌های PC نیز پیاده‌سازی شد.

مزایای این گونه سیستم‌ها عبارتند از:

۱) در این سیستم‌ها، اندازه سخت‌افزار و قیمت آن به نحو چشم‌گیری کاهش یافت.

۲) سیستم‌ها کاربر پسند[1] شدند.

۳) با گسترش این سیستم‌ها، سیستم‌های عامل شبکه‌ای و توزیع‌شده، به‌وجود آمدند.

۶) سیستم‌های توزیع شده[2]

سیستم‌عامل‌های توزیع شده، در یک محیط شبکه‌ای اجرا می‌شوند و نحوه کار در آن‌ها بدین شکل است که بخش‌های مختلف برنامه کاربر، بدون آن‌که خود او متوجه شود، می‌توانند همزمان در چند کامپیوتر مجزا اجرا شده و سپس نتایج نهایی به کامپیوتر اصلی ارسال شود. نکته‌ای که در این بین وجود دارد، این است که کاربران نباید از این امر آگاه شوند که برنامه آن‌ها در کجا به اجرا در می‌آید و یا فایل‌های آن‌ها در کجا قرار دارد. همه این امور باید توسط سیستم‌عامل و به صورت خودکار و با کارایی بالا انجام شود. به‌عبارتی دیگر، می‌توان این‌گونه تصور نمود که سیستم باید از دیدگاه کاربر شفاف[3] یا نامریی باشد. یعنی هر چیز را با نام آن فراخوانی کرده و نیاز به آدرس آن نداشته باشد.

سیستم‌های توزیع شده در دو دسته زیر قرار می‌گیرند:

۱- سیستم‌های با اتصال ضعیف[4]

در این سیستم‌ها تعدادی پردازنده با خطوط ارتباطی مناسب وجود دارند و هر پردازنده دارای پالس ساعت و حافظه مستقل است. از مشخصه‌های این سیستم، سرعت اجرای پایین آن است.

۲- سیستم‌های با اتصال محکم[5]

در این مدل، پردازنده‌ها دارای پالس ساعت یکسان و حافظه مشترک هستند. از مشخصه‌های این سیستم‌ها، سرعت اجرای بالا و پیچیدگی کار آن‌ها می‌باشد.

---

[1] User-friendly
[2] Distributed systems
[3] Transparent
[4] Loosely Coupled System
[5] Tightly Coupled System

مزایای این گونه سیستم‌ها عبارتند از:

۱) یک برنامه می‌تواند به‌طور همزمان بر روی چند کامپیوتر اجرا شود که در نتیجه سرعت انجام محاسبات افزایش می‌یابد. این ویژگی، یکی از مزایای مهم سیستم‌های توزیع شده است.

۲) امکان تسهیم[1] کلیه منابع کامپیوترهای مختلف وجود دارد.

۳) این سیستم‌ها، Functionality (توان انجام انواع عملیات) بالایی دارند.

۴) قابلیت اعتماد (Reliability) در این سیستم‌ها بالاست. به‌عبارتی اگر کامپیوتری در سیستم توزیع شده خراب شود، دیگر کامپیوترها می‌توانند کار را ادامه دهند.

۵) در این سیستم‌ها، ارتباطات از قبیل پست الکترونیک و انتقال فایل‌ها امکان‌پذیر است.

معایب این گونه سیستم‌ها عبارتند از:

۱) این نوع سیستم‌ها به پروتکل شبکه نیاز دارند و با دستگاه واسط مانند اداپتور شبکه و نرم‌افزار کنترل آن درگیر هستند.

۲) باید از نرم‌افزاری برای بسته‌بندی داده تحت پروتکل‌های استفاده شده و بازکردن بسته‌ها استفاده شود.

۷) سیستم‌های موازی[2]

چنین سیستم‌هایی بیش از یک پردازنده دارند که این پردازنده‌ها دارای پالس ساعت (clock) یکسان و حافظه مشترک هستند. سیستم‌های موازی به دو دسته زیر تقسیم می‌شوند:

نامتقارن[3]: در این سیستم، یک پردازنده اجرای سیستم‌عامل را بر عهده داشته و سایر پردازنده‌ها به اجرای برنامه‌های کاربر می‌پردازند.

متقارن[4]: در سیستم‌های چندپردازنده‌ای متقارن، سیستم‌عامل می‌تواند روی هر یک از پردازنده‌های آزاد و یا روی تمام آن‌ها همزمان اجرا شود.

معایب سیستم نامتقارن نسبت به متقارن در این است که، در روش نامتقارن اگر پردازنده‌ای که سیستم‌عامل را اجرا می‌کند، از کار بیفتد، کل سیستم خراب می‌شود و همچنین از آن جا که سیستم‌عامل فقط بر روی یک پردازنده اجرا می‌گردد، امکان کند شدن سرعت و همچنین عدم وجود تعادل بار (balancing) در سیستم وجود دارد. از طرفی مزیت سیستم نامتقارن در این می‌باشد که ساخت آن نسبتا ساده است و از تعمیم سیستم تک‌پردازنده‌ای به‌دست می‌آید.

مزایای این گونه سیستم‌ها عبارتند از:

---

[1] Multiplexing
[2] Parallel system (Multi Processor)
[3] Asymmetric multiprocessors (ASMP)
[4] Symmetric multiprocessors (SMP)

۱) توان عملیاتی بالا: یعنی تعداد کار بیشتری در واحد زمان انجام می‌گیرد. لازم به‌ذکر است که با n برابر کردن تعداد پردازنده‌ها ضریب تسریع n برابر نمی‌شود. چرا که در اکثر مواقع، پیش‌نیاز اجرای برخی فرایندها در سیستم، اجرای سایر فرایندها و استفاده از خروجی آن‌ها است.

۲) صرفه‌جویی اقتصادی: به اشتراک گذاشتن منابعی از قبیل دیسک‌ها، حافظه‌ها برای یک یا چند پردازنده امکان‌پذیر است.

۳) افزایش قابلیت اعتماد: با از بین رفتن و یا نقص در یک پردازنده، اغلب کار سیستم نمی‌خوابد، در حالی‌که در تک پردازنده‌ای این‌گونه نیست. به این مزیت تحمل‌پذیری در برابر خطا[1] می‌گویند.

۱-۲-۲ بررسی سیستم‌عامل‌ها از لحاظ کاربرد صنعتی

این سیستم‌ها در دو دسته زیر قرار می‌گیرند:

۱) سیستم‌عامل‌های بلادرنگ[2]

در سیستم‌های بلادرنگ، هر فرایند دارای مهلت زمانی خاصی برای انجام می‌باشد. زمان پاسخ در این سیستم‌ها باید سریع و تضمین شده باشد، ولی همان‌طور که ملاحظه شد در سیستم‌های اشتراک زمانی، زمان پاسخ مطلوب است ولی اجباری نیست (در مورد سیستم‌های دسته‌ای هیچ محدودیت زمانی در نظر گرفته نمی‌شود). سیستم‌های بلادرنگ و اشتراک زمانی با یکدیگر تناقض دارند و اصولا نمی‌توانند هر دو همزمان در یک سیستم وجود داشته باشند.

انواع سیستم‌عامل بلادرنگ

بلادرنگ سخت[3]

در این سیستم‌ها، کار باید حتما در زمان تعیین شده تمام شود، در غیر این‌صورت خطای غیر قابل برگشتی به سیستم وارد می‌شود. مانند: ربات‌های صنعتی، سیستم کنترل پرواز و MRI یا پرتونگاری پزشکی.

این سیستم‌ها، تضمین می‌کنند که کارهای بحرانی به موقع انجام شوند و معمولا از حافظه ROM استفاده می‌کنند که با قطع جریان برق، اطلاعات از بین نمی‌رود.

بلادرنگ نرم[4]

در این سیستم‌ها، رعایت مهلت زمانی مطلوب است ولی اجباری نیست. به‌عبارتی سیستم سعی خود را می‌کند، ولی اجباری در انجام کار در مهلت زمانی خاص وجود ندارد. مانند: سیستم‌های زمان‌بندی پروازها، زمان‌بندی پروژه، multimedia (صوت و تصویر) در سیستم‌عامل windows و غیره.

---

[1] Fault tolerant
[2] Real time
[3] Hard real time systems
[4] Soft real time systems

از ویژگی‌های این روش آن است که برخی کارها دارای اولویت بالا هستند، و اصولا این سیستم‌ها قابل ترکیب با سیستم‌های دیگر نیز می‌باشند. لازم به‌ذکر است که در این نوع، احتمال خطا یا تاخیر (برخلاف بلادرنگ سخت) وجود دارد.

۲) سیستم‌های تعبیه شده[1]

عموما این سیستم‌ها به این منظور ایجاد می‌شوند تا تعداد مشخصی از وظایف معین را در داخل یک وسیله مدیریت کنند و به نوعی برای کنترل وسایلی مانند وسایل خانگی و ماشین‌ها استفاده می‌شوند. محل قرارگیری یک سیستم تعبیه شده اغلب تراشه‌ای کوچک یا یک بورد است. این سیستم‌ها معمولا مد هسته ندارند و اجرای دستورات در بعضی از آن‌ها به‌صورت بلادرنگ است.

۱-۲-۳ بررسی سیستم‌عامل‌ها از لحاظ ساختاری

در ادامه چند ساختار سیستم‌عامل از قبیل: ۱) ساختار یکپارچه ۲) لایه‌ای ۳) ماشین مجازی ۴) Exokernel ۵) ساختار مشتری-کارگزار (ریز هسته)، که در عمل مورد آزمایش قرار گرفته‌اند، بررسی می‌شود.

۱) ساختار یکپارچه[2]

در این‌گونه پیاده‌سازی، از یک طرف به دلیل بهره‌وری از حداقل فضا و حداکثر کارایی و از طرف دیگر به دلیل محدودیت سخت‌افزار، هیچ نوع عملکرد تقسیم‌بندی ماژولار (پیمانه‌ای) برای سیستم‌عامل وجود نداشته و دسترسی به تمام قسمت‌ها در آن آزاد است. مانند سیستم‌عامل DOS.

در این روش می‌توان این‌گونه در نظر گرفت که سیستم‌عامل به صورت یک مجموعه از رویه‌ها نوشته شده که هر یک از آن‌ها می‌تواند دیگری را به هنگام نیاز فراخوانی کند، اما هیچ امکانی برای مخفی کردن اطلاعات وجود نداشته و از آن‌جا که دسترسی به هر رویه‌ای امکان‌پذیر است، حفاظت وجود ندارد. در سیستم‌های یکپارچه نیز امکان داشتن حداقل یک ساختار کوچک وجود دارد. برای تقاضای سرویس‌های فراهم شده توسط سیستم‌عامل (فراخوان‌های سیستمی)، ابتدا پارامترهای مربوطه را در مکان‌های دقیقا تعریف شده مانند ثبات‌ها یا پشته‌ها قرار می‌گیرند و سپس یک دستورالعمل تله[3] مخصوص اجرا می‌شود که به فراخوان هسته[4] معروف است. این دستورالعمل ماشین را از مد کاربر به مد هسته تغییر حالت داده و کنترل را در اختیار سیستم‌عامل قرار می‌دهد.

۲) ساختار لایه‌ای[5]

در ساختار لایه‌ای، سیستم‌عامل به تعدادی سطح یا لایه تقسیم می‌شود. تقسیم‌بندی لایه‌ها به گونه‌ای است که هر لایه فقط از توابع و سرویس‌های لایه پایین‌تر استفاده می‌کند. به این صورت هر لایه را می‌توان مستقل از لایه‌های دیگر طراحی و خطایابی کرد. لازم به ذکر است که عملکرد این ساختار به روش پیمانه‌ای[1] می‌باشد.

---

[1] Embedded systems
[2] Monolithic
[3] Trap instruction
[4] Kernel call
[5] Layered

مشکلات این روش عبارتند از: تعریف دقیق لایه‌ها، این‌که هر کاری در چه لایه‌ای باشد، کاهش کارایی به این دلیل که یک درخواست برحسب تعداد لایه‌هایی که باید طی کند سربار[2] ایجاد می‌کند.

اولین سیستمی که به این روش ایجاد شد، سیستم THE بود که توسط دکسترا و دانشجویانش در سال ۱۹۶۸ ساخته شد. این سیستم ۶ لایه به شرح زیر دارد:

لایه صفر: تعیین می‌کند CPU در هر لحظه در اختیار کدام فرایند باشد.

لایه یک: مدیریت حافظه اصلی و جانبی را برعهده دارد.

لایه دو: ارتباط هر فرایند و کنسول اپراتور را فراهم می‌کند.

لایه سه: مدیریت دستگاه‌های ورودی و خروجی و بافرکردن اطلاعات آن‌ها را برعهده دارد.

لایه چهار: برنامه‌های کاربران را اجرا می‌کند.

لایه پنج: در این لایه فرایند اپراتور سیستم قرار دارد.

لایه	وظیفه
۵	اپراتور
٤	برنامه های کاربر
۳	مدیریت ورودی و خروجی
۲	ارتباط فرایند - اپراتور
۱	مدیریت حافظه اصلی و جانبی
۰	تخصیص پردازنده و چند برنامگی

شکل ۱-۴: ساختار لایه‌ای THE

۳) ماشین مجازی

این سیستم‌عامل به محض نصب بر روی یک سخت‌افزار، آن‌را شبیه‌سازی می‌کند، به طوری که بتوان سیستم-عامل‌های مختلف دیگری را به طور همزمان روی سیستم‌عامل ماشین مجازی نصب کرد و از آن استفاده نمود. به عبارتی در این ساختار می‌توان تصور نمود که تعدادی ماشین وجود دارد و هر کاربر یک برنامه CMS[3] مخصوص که یک سیستم‌عامل تک کاربره محاوره‌ای می‌باشد را دارد.

از مزایای این روش آن است که هر ماشین مجازی از سایر ماشین‌ها کاملا جدا است، بنابراین هیچ مشکل امنیتی وجود ندارد و برنامه‌های کاربران تداخلی با هم ندارند.

[1] Modularity
[2] Over head
[3] Conversational Monitor System

ایده ماشین مجازی امروزه به وفور در زمینه‌های مختلف مورد استفاده قرار می‌گیرد: مثل اجـرای برنامـه‌هـای Ms-Dos قدیمی بر روی یک پنتیوم. هنگام طراحی پنتیوم و نرم‌افزارهایش، اینتل و مایکروسـافت متوجـه شـدند کـه بـا تقاضاهای زیادی در مورد اجرای نرم‌افزارهای قدیمی بر روی سخت‌افزار جدید مواجه خواهند شد. به همـین دلیـل، اینتل یک مود ۸۰۸۶ مجازی را بر روی پنتیوم تهیه کرد. در این مود ماشین شبیه ۸۰۸۶ (از لحاظ نـرم‌افـزاری) عمـل می‌کند و از آدرس‌دهی ۱۶ بیتی با محدوده یک مگابایتی برخوردار است.

مثالی دیگر از ایده ماشین مجازی، در شرکت‌هایی اسـت کـه خـدمات میزبـانی وب[1] را ارایـه مـی‌کننـد. در ایـن شرکت‌ها، عموما برنامه‌های وب مختلف و با نیاز به سیستم‌های عامل مختلـف، مـی‌خواهنـد بـر روی یـک سـرور نصب شوند. VMWare و نیز Virtual PC مایکروسافت برای چنین اهدافی عرضه شدند..

حوزه دیگر کاربرد ماشین‌های مجازی، البته با اندکی تفاوت، اجرای برنامه‌های جاوا اسـت. هنگـامی کـه شـرکت Sun Microsystem زبان برنامه‌نویسی جاوا را به وجود آورد، یک ماشین مجازی به نام JVM [2] ابـداع نمـود. متـرجم جاوا کدها را برای JVM تولید می‌کند که معمولا به وسیله نرم‌افزار مفسر JVM اجرا می‌شود. مزیـت ایـن روش، آن است که کد JVM می‌تواند بر روی اینترنت حرکت کند و بر روی هر نوع کامپیوتری که مفسر JVM داشـته باشـد، اجرا شود.

۴) Exokernels

این ساختار با انجام تغییراتی در روش ماشین مجازی به وجود آمد. به این صورت کـه در پـایین‌تـرین لایـه، یـک برنامه به نام exokernel در مد هسته اجرا می‌شود و کارش این است که منابع را به ماشـین‌هـای مجـازی تخصـیص دهد. سپس استفاده آن‌ها را از منابع مورد بررسی قرار می‌دهد تا مطمئن شود هیچ یک از ماشین‌ها سعی نـدارد کـه از منابع دیگری استفاده کند. هر ماشین مجازی سطح کاربر می‌تواند سیستم‌عامل خودش را بـه اجـرا در آورد، فقـط بـا این تفاوت که در این‌جا هر کدام از آن‌ها محدودند که تنها از منابعی استفاده نمایند که آن را در خواسـت کـرده و بـه آن‌ها تخصیص داده شده است.

فایده دیگر طرح exokernel این است که دیگر نیازی به لایه نگاشت[3] نیست. (در طراحی‌های دیگـر، هـر ماشـین مجازی فکر می‌کند که مثلا منبع دیسک خودش را با بلوک‌هـایی از صـفر تـا یـک مقـدار حـداکثر در اختیـار دارد، بنابراین مانیتور ماشین مجازی مجبور است که جدولی را نگه دارد تـا آدرس‌هـای مربـوط بـه دیسـک را دوبـاره بـه آدرس‌های واقعی در دیسک نگاشت نماید. اما exokernel نیاز به این نگاشت مجـدد نـدارد و فقـط بایـد بدانـد کـه کدام منبع به کدام ماشین تخصیص داده شده است.)

[1] Web-hosting
[2] Java Virtual Machine
[3] Mapping

۵) ساختار مشتری - خدمت‌گزار[1] (ریز هسته[2])

ایده اصلی در ساختار مشتری -خدمت‌گزار، این است که تاحد ممکن کدها را به لایه‌های بالاتر منتقل و از سیستم‌عامل حذف نماییم، تا نهایتا یک هسته کمینه داشته باشیم. از آن‌جا که در این معماری یک هسته کوچک خواهیم داشت، به آن ریزهسته نیز اطلاق می‌شود.

روش به این صورت است که اکثر وظایف سیستم‌عامل را در سطح فرایندهای کاربر پیاده‌سازی می‌کنیم. برای درخواست یک سرویس، مانند خواندن یک بلوک از فایل، فرایند کاربر (که اکنون به عنوان فرایند مشتری شناخته می‌شود) یک درخواست به فرایند خدمت‌گزار ارسال می‌نماید و از آن می‌خواهد که کارش را انجام دهد و پاسخ را برگرداند. کار هسته در این مدل برقراری ارتباط بین مشتری‌ها و خدمت‌گزارها است. البته هسته وظایف مهم دیگری نیز دارد. به طور کلی وظایف هسته در این مدل عبارتند از: ۱) تبادل پیام بین مشتری‌ها و خدمت‌گزارها ۲) زمان‌بندی و ایجاد چندبرنامگی ۳) بخش سطح پایین مدیریت حافظه مانند برنامه‌ریزی ثبات‌های سخت‌افزار مدیریت حافظه ۴) بخش مدیریت سطح پایین I/O که برنامه‌ریزی ثبات‌های ویژه کنترل‌کننده‌ها و اموری را بر عهده دارند که اگر در سطح کاربر ایجاد شوند، امنیت سیستم را خدشه‌دار خواهند ساخت. لازم به ذکر است که دیگر خدمات سیستم‌عامل به خدمت‌گزار داده می‌شود. این خدمت‌گزارها در مد کاربر اجرا شده و ریزهسته با آن‌ها مانند کاربردهای دیگر رفتار می‌کند.

در این روش سیستم‌عامل به چندین بخش تقسیم شده است که هر یک از آن‌ها فقط یکی از وجوه سیستم‌عامل مانند خدمات فایل، خدمات فرایند، خدمات ترمینال و خدمات حافظه را به دست می‌گیرد و در نتیجه هر بخش را می‌توان کوچک و قابل مدیریت طراحی کرد. همان‌طور که گفته شد اجرای کلیه فرایندهای خدمت‌گزار در مد کاربر انجام می‌شوند (نه در مد هسته) و هیچ‌کدام از آن‌ها دسترسی مستقیم به سخت‌افزار ندارند. در نتیجه اگر یک اشکال مثلا در خدمت‌گزار فایل ایجاد شود، فقط موجب فروپاشی خدمات فایل خواهد شد و معمولا موجب خوابیدن کل سیستم نمی‌شود.

در کل می‌توان مزایای این ساختار را به‌شرح زیر دانست:

الف) سادگی طراحی و پیاده‌سازی سیستم‌عامل، به‌دلیل تقسیم آن به بخش‌های مختلف.

ب) کاهش احتمال خرابی کل سیستم به دلیل اجرای اغلب فرایندها در مد کاربر و همچنین این نکته که با خراب شدن یک سرویس، فقط آن سرویس غیرفعال شده و کل سیستم از کار نمی‌افتد.

ج) سازگاری این ساختار با سیستم‌های توزیع شده.

---

[1] Client/Server
[2] Micro Kernel

۱-۳ نگاه کلی به سخت‌افزار و مکانیزم کار I/O

یک سیستم‌کامپیوتری شامل یک یا چند نمونه از سخت‌افزارهای زیر است که اتصال آن‌ها با یکدیگر امکان اجرای برنامه‌ها را فراهم می‌کنند:

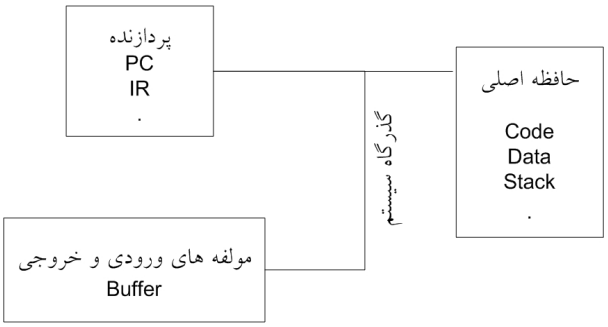
۱- پردازنده: این عنصر ضمن انجام پردازش داده‌ها (ALU)، عملیات کامپیوتر را کنترل (CU) می‌کند که معمولا به آن واحد پردازش مرکزی[1] می‌گویند.

۲- حافظه اصلی (RAM – حافظه دسترسی تصادفی[2]): داده‌ها و برنامه‌ها را ذخیره می‌کند که حافظه از نوع ناپایدار است.

۳- مولفه‌های ورودی و خروجی: داده‌ها را بین کامپیوتر (حافظه اصلی و پردازنده) و محیط خارجی (حافظه جانبی، پایانه‌ها و غیره) منتقل می‌کنند.

۴- گذرگاه: واسط ارتباطی بین پردازنده، حافظه اصلی و مولفه‌های ورودی و خروجی است.

شکل زیر این اجزا و ارتباط آن‌ها را نشان می‌دهد:



شکل ۱-۵: عناصر اصلی سخت افزار یک سیستم کامپیوتری

۱-۳-۱ ثبات‌های پردازنده

همان‌طور که می‌دانیم، واحد پردازش مرکزی از یک مجموعه ثبات برای اجرای دستورات و پردازش داده‌های ذخیره شده در حافظه اصلی، استفاده می‌کند. ثبات‌ها را می‌توان بر حسب وظیفه‌ای که دارند در دو گروه زیر تقسیم‌بندی کرد:

الف) ثبات‌های قابل رویت برای کاربر (Visible):

این نوع ثبات‌ها، قابل رویت توسط برنامه کاربردی یا کاربر هستند، که در زبان ماشین یا اسمبلی، برنامه‌ساز مستقیما و در زبان‌های سطح بالا، کامپایلر به‌طور هوشمند به آن‌ها دسترسی دارد. لازم به‌ذکر است که در برخی از

[1] CPU (Centeral Processing Unit)
[2] Random Access Memory

زبان‌ها، که سطح میانی هستند مانند زبان C، برنامه‌ساز نیز می‌تواند مستقیما به آن دسترسی داشته باشد. مانند ثبات-های DR, AR.

ب) ثبات‌های کنترل و وضعیت (غیر قابل رویت Invisible):

این نوع ثبات‌ها برای کنترل عملیات پردازنده و همچنین رویه‌های ممتاز سیستم‌عامل برای کنترل اجرای برنامه‌ها، در نظر گرفته شده‌اند. این ثبات‌ها توسط کاربر قابل دسترسی نیستند و فقط سیستم‌عامل به آن‌ها دسترسی دارد. مانند ثبات‌های IR, PC و مجموعه ثبات‌های PSW.

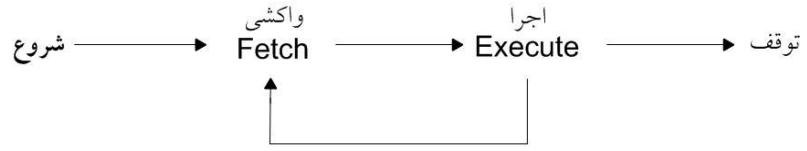
ثبات PSW[1]

پردازنده‌ها معمولا شامل یک یا چند ثبات هستند که تحت عنوان کلمه وضعیت برنامه (PSW) شناخته می‌شوند. بیت‌های وضعیت (پرچم‌ها) در مراحل مختلف اجرای یک برنامه در این ثبات(ها) ذخیره می‌شوند. کدهای وضعیت معمولا خود شامل بیت‌هایی هستند که به عنوان نتیجه عملیات توسط سخت‌افزار مقدارگذاری می‌شوند. بطور مثال در یک عملیات محاسباتی ممکن است نتیجه مثبت، منفی و یا سرریز باشد. علاوه بر کدهای وضعیت، بیت‌های فعال/غیرفعال کردن (یا کنترل) وقفه، بیت حالت (مد) کاربر یا ناظر (مد هسته) بودن پردازنده نیز در این ثبات(ها) تعریف می‌شود.

نکته قابل ذکر این است که، پردازنده به ثبات‌های مهم، از جمله ثبات شمارنده برنامه (PC) و PSW یک توجه خاص دارد و در هنگام وقفه آن‌ها را در پشته ذخیره می‌کند تا آدرس برگشت و اطلاعات مهم فرایند در حال اجرا را از دست ندهد. اما ثبات اشاره‌گر پشته (SP) و سایر ثبات‌ها مانند ثبات‌های داده از متعلقات اصلی فرایند محسوب می‌شوند.

۱-۳-۲ اجرای دستورالعمل‌ها

همان‌طور که می‌دانیم، هر برنامه‌ای که بخواهد اجرا شود شامل تعدادی دستورالعمل است که درحافظه ذخیره شده است و CPU برای اجرای برنامه لازم است آن‌ها را از حافظه بخواند. فرایندی که برای پردازش یک دستورالعمل لازم است، مطابق شکل زیر، چرخه واکشی و اجرا نامیده می‌شود:



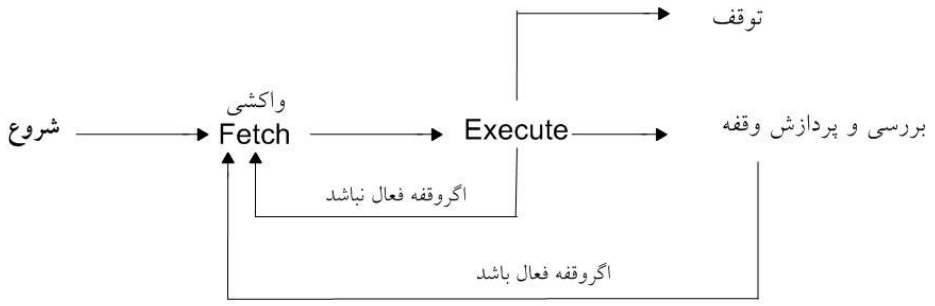
شکل ۱-۶: چرخه اصلی دستورالعمل

۱-۳-۳ عملکرد سیستم‌عامل در کار با وقفه‌ها

با توجه به اختلاف سرعت پردازنده با دستگاه‌های ورودی و خروجی، سخت افزار به همراه سیستم‌عامل باید روشی جهت هماهنگ کردن کار این دو بخش به وجود آورند. حتی در مورد دستگاه‌های سریع نیز، با توجه به

[1] Program Status Word

مجزا بودن این دو بخش، هماهنگی آن‌ها لازم است، که از روش‌های مفید و مورد استفاده در این زمینه وقفه[1] می‌باشد.



شکل ۱-۷: وقفه‌ها و عملکرد سیستم‌عامل در کار با دستگاه‌های ورودی و خروجی

همان‌طور که در شکل ۱-۷ ملاحظه می‌شود، برای حمایت از وقفه یک چرخه، به چرخه دستورالعمل اضافه می‌شود. در چرخه وقفه، پردازنده بروز وقفه را بررسی می‌کند. اگر هیچ وقفه‌ای مطرح نباشد، پردازنده برای واکشی جلو رفته و دستورالعمل بعدی را از برنامه جاری واکشی می‌کند و چنان‌چه وقفه‌ای مطرح باشد، پردازنده اجرای برنامه جاری را مسکوت گذاشته و پس از ذخیره‌سازی ثبات‌ها در پشته، رویه سرویس وقفه[2] مربوطه را اجرا می‌کند. می‌توان این‌طور تصور نمود که منظور از وقفه، انتقال کنترل برنامه، از برنامه‌ی جاری به برنامه‌ی دیگر به‌نام رویه سرویس وقفه است، که پس از تقاضای داخلی (یا خارجی) صورت گرفته و پس از اجرای سرویس‌دهی وقفه، کنترل مجددا به برنامه اصلی باز می‌گردد.

وقفه‌ها معمولا، دارای یک سطح اولویت[3] سخت‌افزاری هستند. پردازنده نیز اولویت خاص خود را دارد و فقط وقفه‌های با سطح اولویت بالاتر از پردازنده، پردازش می‌شوند و وقفه‌های با اولویت پایین‌تر تا زمانی که پردازنده اولویت خود را پایین نیاورد، پردازش نشده باقی می‌مانند. سطح اولویت پردازنده در PSW ذخیره می‌شود که می‌توان آنرا با تغییر بیت‌های متناظر در PSW، تغییر داد.

انواع وقفه

وقفه‌ها به دو نوع سخت‌افزاری و نرم‌افزاری تقسیم می‌شوند که در ادامه بررسی می‌شوند:

۱) وقفه‌های سخت‌افزاری (خارجی)

از سوی یک سخت‌افزار خارج از پردازنده (و یا داخل پردازنده) و به صورت یک سیگنال ناهمگام (در یک لحظه تصادفی که از قبل مشخص نیست در کجای برنامه جاری به وقوع می‌پیوندد) به پردازنده ارسال می‌شود. این نوع وقفه‌ها به چند دسته زیر تقسیم می‌شوند:

---

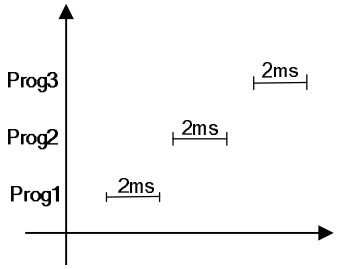
[1] Interrupt
[2] ISR (Interrupt Service Routine)
[3] Priority level

وقفه ورودی و خروجی

این وقفه توسط کنترل کننده دستگاه‌های ورودی و خروجی، در هنگام اتمام عملیات و یا در هنگام ایجاد خطا در عمل I/O رخ می‌دهد. مانند وقفه صفحه کلید.

وقفه زمان‌سنج[1] (وقفه ساعت)

Timerها یا زمان‌سنج‌های داخلی پردازنده، به منظور تعیین زمان اجرای پردازنده در هر برهه‌ی زمانی[2] استفاده می‌شوند. در سیستم‌های اشتراک زمانی، همانند شکل ۱-۸ سیستم‌عامل یک Timer را تنظیم کرده و ضمن سپردن پردازنده به فرایند، خودش کاملاً کنار می‌رود. پس از طی مدت زمان تعیین شده در Timer، یک وقفه تولید می‌شود که در نتیجه آن پردازنده به سیستم‌عامل باز می‌گردد. همچنین در سیستم‌عامل بعضی از اعمال، نظیر تست حافظه، چک کردن سخت افزار و غیره باید به‌طور مرتب در فواصل زمانی تعیین شده انجام شوند، که در مجموع برای انجام این نوع کارها، سیستم‌عامل از ساعت داخلی ماشین کمک می‌گیرد.



شکل ۱-۸: نحوه اجرا در سیستم‌های اشتراک زمانی

خطای ماشین (نقص سخت‌افزاری)

مانند خطای بیت توازن در حافظه اصلی (RAM)

وقفه Restart

۲) وقفه‌های نرم‌افزاری (داخلی)

این وقفه‌ها به‌صورت همگام و در اثر اجرای دستورالعمل خاصی از برنامه جاری، به وقوع می‌پیوندد. وقفه‌های نرم‌افزاری به چند دسته زیر تقسیم می‌شوند:

فراخوان‌های سیستمی (API یا SVC)

این وقفه‌ها عموما در حالتی واقع می‌شوند که برنامه‌ی کاربر نیاز به استفاده از مد ناظر و امکانات آن‌را داشته باشد. مثلا، در یک برنامه به‌جای انجام کارهای پیچیده با I/O برای خواندن و ارسال داده‌ها، می‌توان آن‌را توسط وقفه (فراخوان‌های سیستمی) به ناظر واگذار کرد تا کار با اطلاعات دقیق‌تر انجام گیرد. مثال دیگر، دستور exit در انتهای یک برنامه است که وقتی اجرا به این خط می‌رسد، هسته سیستم‌عامل فراخوانی شده و اعمال لازم برای خروج فرایند انجام می‌گیرد.

---

[1] Timer
[2] Time slice

سیگنال‌ها

سیگنال، وقفه‌ای مجازی است که توسط یک فرایند، سیستم‌عامل و یا کاربر، به یک یا چند فرایند ارسال می‌شود. فرایندی که سیگنال را دریافت می‌کند، می‌تواند از آن صرف‌نظر کند و یا با اجرای یک تابع خاص به سیگنال پاسخ دهد.

خطای برنامه (استثنا – تله)

مانند خطای سرریز، تقسیم بر صفر، نقص‌های حفاظتی از قبیل اجرای یک دستورالعمل ممتاز در مد کاربر و مراجعه به یک آدرس نادرست یا غیر مجاز در حافظه. لازم به‌ذکر است که در برخی از متون سیستم‌عامل به این نوع خاص از وقفه، **تله**[1] می‌گویند و در برخی دیگر کلیه وقفه‌های نرم‌افزاری را تله می‌نامند. اصطلاح تله، حاکی از این است که برنامه خاطی توسط سخت‌افزار حفاظت در تله هسته گرفتار می‌شود.

بررسی عملکرد دو حالته[2]

سیستم‌عامل خود یک برنامه است، با این تفاوت که اختیارات بیش‌تری نسبت به سایر برنامه‌ها دارد. برای فراهم کردن این اختیارات ویژه به‌گونه‌ای که سایر برنامه‌های کاربردی مجاز به داشتن آن‌ها نباشند، سخت‌افزار باید مدهای (حالت‌های) مختلفی از اجرا را پشتیبانی کند. لذا دو مد اجرای هسته[3] (راهبری)[4] و کاربر[5] مورد استفاده قرار می‌گیرند و در یک بیت ثبات PSW هر لحظه مد اجرای سیستم ذخیره می‌شود. در نتیجه تلاش برنامه‌های عادی، برای اجرای فعالیت‌هایی که فقط برای سیستم‌عامل رزرو شده‌اند، (نظیر دسترسی به وسایل I/O، ثبات‌های اصلی و غیره) هنگامی که در مد کاربر قرار داریم، باعث بروز تله خواهد شد.

لازم به‌ذکر است که در ابتدای کار و با Restart کامپیوتر، بیت مد، صفر است و سیستم‌عامل قبل از این‌که کنترل را به برنامه بدهد، آن را یک می‌کند و از آن‌جا که هر وقفه، کنترل را به سیستم‌عامل بر می‌گرداند، با تولید آن، بیتِ مد مجددا صفر می‌شود.

۱-۳-۴ ارتباط دستگاه‌های ورودی و خروجی با پردازنده، حافظه و سیستم‌عامل

انتقال داده‌ها در یک سیستم‌کامپیوتری از طریق گذرگاه صورت می‌پذیرد. برای این منظور، لازم است چند عمل زیر انجام گیرد تا داده‌ها از دستگاه به فضای آدرس حافظه برنامه کاربر منتقل شوند:

۱) تقاضای ورودی و خروجی کاربر به فرمان دستگاه تبدیل شده و به آن ارسال گردد.

۲) مکانیزمی برای انتقال داده از حافظه یا فرستادن آن به حافظه ایجاد شود.

در راستای انجام اعمال فوق، سیستم‌عامل باید به‌عنوان رابط سخت‌افزار و برنامه‌ای که درخواست ورودی و خروجی کرده است و همچنین مدیر دستگاه‌های I/O نقش ایفا کند. در واقع قرار گرفتن سیستم‌عامل در این دو نقش، از سه ویژگی زیر در دستگاه‌های ورودی و خروجی ناشی می‌شود:

---

[1] Trap
[2] Dual mode
[3] Kernel mode
[4] Supervisor mode
[5] User mode

۱) از آن‌جا که معمولا برنامه‌های متعددی به طور همزمان در حال اجرا هستند، ممکن است سیستم I/O بـه اشـتراک گذاشته شود.

۲) سیستم‌های I/O برای انتقال اطلاعات اغلب از وقفه‌ها استفاده می‌کنند. همان‌طور که دیده‌ایم، وقفه‌ها باعـث انتقـال به مد هسته یا ناظر می‌شوند که در نتیجه اداره کردن آن‌ها وظیفه سیستم‌عامل است.

۳) کنترل سطح پایین دستگاه‌های I/O معمولا کار پیچیده‌ای است.

با توجه به ویژگی‌های گفته شده در سیستم‌های I/O، وظایف سیستم‌عامل در این زمینه را می‌تـوان در مـوارد زیـر خلاصه کرد:

۱) سیستم‌عامل تضمین می‌کند که برنامه‌های کاربر، فقط به بخش‌های مجاز دسترسی داشته باشند. بـه طـور مثـال سیستم‌عامل نباید اجازه خواندن یا نوشتن یک فایل را بـه برنامـه‌ای بدهـد کـه صـاحب فایـل بـه آن برنامـه اجـازه دسترسی را نداده باشد. برای این منظور در یک سیستم با دستگاه‌های I/O مشترک، بایـد از اسـتفاده مسـتقیم برنامـه-های کاربر از دستگاه‌های I/O ،جلوگیری به‌عمل آید.

۲) سیستم‌عامل با نگه‌داری روال‌هایی برای اداره نمودن سطح پایین دستگاه I/O سبب ایجاد یک محـیط انتزاعـی از آن برای برنامه کاربر می‌شود.

۳) سیستم‌عامل باید وقفه‌های تولید شده از دستگاه‌های I/O را نیز اداره کند.

۴) سیستم‌عامل تلاش می‌کند با زمان‌بندی منابع، امکان دسترسی مساوی و عادلانه برنامه‌های کـاربر بـه منـابع I/O مشترک را ایجاد کند.

سیستم‌عامل برای انجام این وظایف، باید بتواند با دسـتگاه‌هـای I/O ارتبـاط مسـتقیم برقـرار کنـد و از دسترسـی مستقیم برنامه‌های کاربر به دستگاه‌های I/O جلوگیری کند. برای این منظور لازم است سیستم‌عامل بـه دسـتگاه‌هـای I/O فرمان دهد. یک فرمان می‌تواند یک دستور عادی مثل خواندن یا نوشتن باشد و یا اعمـالی کـه خـاص دسـتگاه مربوطه است (مثل عمل جستجو در دیسک). از طرفی لازم است که دسـتگاه، سیسـتم‌عامـل را از انجـام عمـل I/O مطلع کند (مثلا زمانی که دیسک عمل جستجو را به اتمام رساند، سیستم‌عامل را بـا خبـر کنـد) و در نهایـت داده‌هـا بتوانند بین حافظه و دستگاه I/O مبادله شوند (به عنوان مثال بلاک خوانده شده از دیسـک بـه حافظـه منتقـل شـود). برای فهم کامل و دقیق این مکانیزم نگاهی گذرا به سخت‌افزار I/O خواهیم داشت.

اصول سخت‌افزاری I/O

در این‌جا به شرح مختصری از سخت‌افزار I/O در سه قسمت دستگاه‌های I/O، کنترل‌کننـده‌هـای دسـتگاه و I/O نگاشت شده، خواهیم پرداخت.

دستگاه‌های I/O

دستگاه‌های I/O به دو دسته‌ی دستگاه‌های بلوکی[1] و دستگاه‌های کاراکتری[2] تقسیم می‌شوند.

[1] Block device
[2] Character device

دستگاه بلوکی وسیله‌ای است که اطلاعات را در بلوک‌هایی با اندازه معین، که هر کدام با آدرس خودشان مشخص شده‌اند، ذخیره می‌کند. حدود اندازه بلوک‌های معمولی از ۵۱۲ بایت تا ۳۲۷۶۸ بایت می‌باشد.

ویژگی اساسی یک دستگاه بلوکی این است که آدرس‌دهی، خواندن و نوشتن هر بلوک را به طور مستقل از بقیه بلوک‌ها امکان‌پذیر می‌سازد. دیسک‌ها متداول‌ترین دستگاه از این نوع هستند. از طرفی دستگاه کاراکتری، یک جریان از کاراکترها را بدون توجه به هیچ ساختار بلوکی دریافت نموده و یا تحویل می‌دهد و بنابراین قابلیت آدرس‌دهی و جستجو در آنها وجود ندارد. به طور مثال چاپگر و یا ماوس یک نوع دستگاه کاراکتری هستند.

کنترل‌کننده‌های دستگاه

واحدهای I/O از اجزاء مکانیکی و الکترونیکی تشکیل شده‌اند. اجزاء الکترونیکی، وفق دهنده[1] یا کنترل‌کننده[2] دستگاه نامیده می‌شوند. اکثر کنترل‌کننده‌ها می‌توانند دو، چهار و یا حتی هشت دستگاه مشابه و یا یکسان را اداره نمایند. بخش مکانیکی، خود دستگاه است. تفکیک کنترل‌کننده از دستگاه به این دلیل است که سیستم‌عامل تقریبا همیشه با کنترل‌کننده و نه با خود دستگاه سر و کار دارد.

I/O نگاشت شده در حافظه[3]

هر کنترل‌کننده تعدادی ثبات دارد که برای ارتباط با پردازنده به‌کار می‌روند. با نوشتن در این ثبات‌ها، سیستم‌عامل می‌تواند فرمان تحویل داده، دریافت داده، خاموش یا روشن کردن و یا سایر فرمان‌ها را به دستگاه‌ها بدهد. با خواندن از این ثبات‌ها، سیستم‌عامل می‌تواند از وضعیت یک دستگاه اگاه شود.

بسیاری از دستگاه‌ها علاوه بر ثبات‌های کنترلی، دارای یک بافر داده می‌باشند که سیستم‌عامل می‌تواند از آن خوانده و یا در آن بنویسد. چگونگی ارتباط پردازنده با ثبات‌های کنترلی و بافر داده دستگاه‌ها، به عنوان یک مساله اساسی است که به دو روش انجام می‌گیرد: روش اول این‌که به هر ثبات کنترلی یک شماره پورت I/O که یک عدد صحیح ۸ و یا ۱۶ بیتی است، اختصاص داده شود که به این ترتیب پردازنده می‌تواند عمل خواندن و نوشتن ثبات کنترلی را انجام دهد و یا این‌که این ثبات‌ها، در فضای حافظه قرار گیرند که این طرح، I/O نگاشت شده در حافظه نامیده می‌شود. لازم بذکر است که در روش اخیر، به هر ثبات کنترلی یک آدرس یکتای حافظه تخصیص داده شده که به هیچ حافظه دیگری اختصاص نیافته است.

معمولا ثبات‌های کنترل‌کننده، یک یا چند بیت وضعیت دارند که می‌توانند برای تعیین کامل شدن یک عملیات خروجی و یا در دسترس بودن داده‌های جدید یک دستگاه ورودی، تست شوند. پردازنده می‌تواند با اجرای یک حلقه، در هر دور، یک بیت وضعیت را تا زمان آماده شدن دستگاه برای پذیرش و یا فراهم نمودن داده‌های جدید، تست نماید. این عمل **I/O برنامه‌نویسی شده[4] یا سرکشی[5]** نامیده می‌شود. از این روش نمی‌توان همیشه استفاده کرد، چرا که مقداری از زمان پردازنده صرف چک کردن وضعیت دستگاه‌ها می‌شود که ایده‌ال نیست.

---

[1] Adapter
[2] Device Controller
[3] Memory-mapped I/O
[4] I/O programmed
[5] Polling

بسیاری از کنترل‌کننده‌ها آمادگی خود را جهت خواندن و یا نوشتن در ثبات‌هایشان، با استفاده از **وقفه‌ها** بـه اطـلاع پردازنده می‌رسانند. این مکانیزم، که **I/O مبتنی بر وقفه**[1] نامیده می‌شود، مشکل ذکر شده در روش سرکشی را ندارد.

دسترسی مستقیم به حافظه DMA[2]

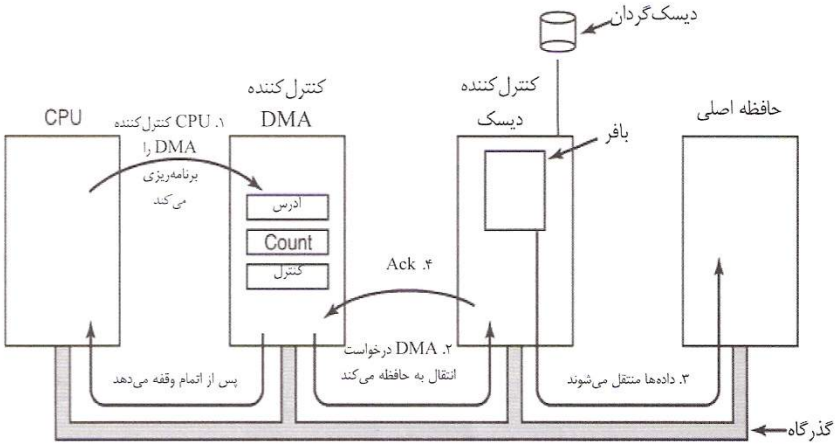
پردازنده در صورت وجود و یا عدم وجود I/O نگاشت شده در حافظه، نیاز به آدرس‌دهی کنترل‌کننده دسـتگاه‌هـا جهت تبادل داده با آن‌ها دارد. پردازنده می‌تواند به کنترل‌کننده دستگاه‌ها درخواستِ داده بـا نـرخ یـک بایـت در هـر دفعه را بدهد، اما چنین عملی برای وسیله‌ای مانند دیسک که بلوک‌های داده زیادی تولید می‌نماید، وقـت پردازنـده را به شدت هدر می‌دهد. در نتیجه از یک روش متفاوت با نام DMA استفاده می‌شود. یـک سیسـتم تنهـا در شـرایطی می‌تواند از DMA استفاده کند که سخت‌افزار کنترل‌کننده DMA داشته باشد. کنترل‌کننده DMA می‌توانـد مسـتقل از پردازنده و بدون توجه به مکان فیزیکی‌اش، به گذرگاه سیستم دسترسی داشته باشد. این کنترل‌کننـده شـامل چنـدین ثبات است که می‌توانند به وسیله پردازنده خوانده یا نوشته شوند. این ثبات‌ها عبارتنـد از یـک ثبـات آدرس حافظـه، یک ثبات شمارش‌گر بایت و یک یا چند ثبات کنترلی. ثبات‌های کنترلی، پورت‌های I/O مورد استفاده جهـت انتقـال (خواندن یا نوشتن از دستگاه I/O)، واحد انتقال (بایت و یا کلمه) و تعداد بایـت‌هـای انتقـال یافتـه در یـک burst را تعیین می‌نمایند. در ادامه چگونگی عملکرد DMA را با یک مثال (خواندن دیسک) بررسی می‌کنیم.

در ابتدا پردازنده کنترل‌کننده DMA را با تنظیم ثبات‌هایش برنامه‌ریزی می‌نماید، به‌گونه‌ای که کنتـرل‌کننـده خواهـد دانست که چه چیزی را باید به کجا منتقل نماید (مرحله ۱ از شکل ۱–۹). همچنین فرمانی را به کنترل‌کننـده دیسـک صادر می‌کند تا داده‌ها را از بافرهای داخلی دیسک خوانده و صحت آن‌ها را بررسـی کنـد (یکـی از اعمـالی کـه در کنترل‌کننده دیسک انجام می‌شود، تشخیص خطا است). هنگامی که داده‌های معتبر درون بـافر کنتـرل‌کننـده دیسـک قرار گیرند، عملیات DMA می‌تواند آغاز شود.

کنترل‌کننده DMA انتقال را با صدور یک درخواست خواندن از طریق گذرگاه، به کنترل‌کننده دیسـک آغـاز مـی‌کند (مرحله ۲). این درخواست خواندن مانند سایر درخواست‌های خواندن بوده و کنترل‌کننده دیسک نمـی‌دانـد کـه آیا درخواست از پردازنده آمده است و یا از کنترل‌کننده DMA. معمولاً آدرس خانـه‌ای از حافظـه کـه بایـد درون آن نوشته شود بر روی خطوط آدرس گذرگاه قرار دارد، به طوری که وقتی کنترل‌کننده دیسک، کلمـه بعـدی را از بـافر درونی‌اش واکشی می‌کند می‌داند کجا باید آن‌را بنویسد. نوشتن در حافظه، یک سیکل استاندارد دیگـر گـذرگاه مـی‌باشد (مرحله ۳). هنگامی که نوشتن کامل می‌شود، کنترل‌کننده دیسک یک سیگنال تصدیق به کنترل‌کننـده DMA از طریق گذرگاه می‌فرسـتد (مرحلـه ۴). سـپس کنتـرل‌کننـده DMA آدرس حافظـه مـورد اسـتفاده را افـزایش داده و شمارش‌گر بایت را کاهش می‌دهد. در صورتی که شمارش‌گر بایت هنوز بیش‌تر از صفر باشد، مراحـل ۲ الـی ۴ تـا صفر شدن آن تکرار می‌شوند. در این مرحله یک وقفه ایجاد می‌نماید. هنگامی که سیستم‌عامل فعال می‌شود، مجبـور نیست که یک بلوک را در حافظه کپی نماید، چون آن بلوک قبلا در حافظه قرار داده شده است.

---

[1] Interrupted I/O
[2] Direct Memory Access



شکل ۱-۹: خطوط عملیات یک انتقال DMA

سوالات فصل اول

۱- فعالیت‌های اصلی یک سیستم‌عامل را در رابطه با مدیریت فرایند ذکر کنید.

ایجاد و حذف فرایندها، زمان‌بندی فرایندها برای دریافت منابع از جمله پردازنده، همزمانی و همگام‌سازی فرایندها، ارتباط بین فرایندها و اداره بن‌بست.

۲- می‌دانیم برای استفاده‌ی بهینه از سخت‌افزار نیازمند سیستم‌عامل هستیم. چه زمانی ممکن است که سیستم-عامل این اصل را نادیده گرفته و به عبارت دیگر از منابع به طور بهینه استفاده نکند؟ چرا چنین سیستمی زیان-آور نیست؟

در سیستم‌های تک کاربره، هدف از وجود سیستم‌عامل راحتی کاربر در استفاده از سیستم می‌باشد. لذا اکثر اوقات ممکن است واسطِ گرافیکی، چرخه‌های عملیاتی پردازنده را برای تعامل بهتر با کاربر هدر دهد. این امر اگرچه باعث هدر رفتن زمان پردازنده می‌شود، اما به نوعی باعث راحتی کار کاربر شده که این اصل مهم‌ترین هدف در این‌گونه سیستم‌ها است.

۳- تعاریف مختلف سیستم‌عامل را در نظر بگیرید. به نظر شما آیا سیستم‌عامل باید شامل کاربردهایی مثل مرورگر وب و برنامه‌های پست الکترونیکی باشد یا خیر؟

کاربردهایی مثل مرورگر وب و برنامه‌های پست الکترونیکی می‌توانند جزیی از سیستم‌عامل باشند تا سبب راحتی دسترسی کاربران به نیازهایشان شوند. منتهی این کار ممکن است سبب کم‌رنگ شدن اهداف اصلی سیستم-عامل شود. با این حال، با افزایش حجم سیستم‌عامل می‌توان این‌گونه کاربردها را نیز پوشش داد.

۴- آیا عمل تمایز بین مد کاربر و مد ناظر، شکلی از سیستم حفاظت (امنیت) است؟ چگونه؟

بله، تمایز بین مد کاربر و ناظر، شکلی از سیستم حفاظت (امنیت) است.

دستورالعمل‌های ویژه تنها زمانی که پردازنده در مد هسته قرار دارد اجرا می‌شوند و به همین ترتیب دستگاه‌های سخت‌افزاری، فقط زمانی می‌توانند مورد دسترسی قرار گیرند که پردازنده در مد هسته باشد. همچنین فعال یا غیرفعال شدن وقفه در حین اجرای یک برنامه، تنها زمانی امکان‌پذیر است که پردازنده در مد هسته باشد. بنابراین پردازنده در مد کاربر توانایی نسبتا محدودی برای اجرای برنامه‌ها دارد و با استفاده از این مکانیزم، در مقابل دسترسی‌های غیر مجاز کاربر، حفاظت به وجود می‌آید.

۵- کدام‌یک از دستورات زیر باید ممتاز باشند (در مد هسته اجرا شوند)؟

۱) مقداردادن به تایمر ۲) خواندن ساعت سیستم ۳) تنظیم ساعت سیستم ۴) پاک کردن حافظه

۵) غیرفعال کردن وقفه ۶) تغییر از مد کاربر به مد ناظر ۷) دسترسی به وسایل ورودی/خروجی

۸) مقداردهی مدخل‌های جدول حالت وسایل ۹) تغییر ثبات‌های مدیریت حافظه

۱۰) نوشتن در شمارنده برنامه ۱۱) تغییر اولویت پردازنده ۱۲) راه‌اندازی مجدد کامپیوتر

۱۳) خواندن PSW ۱۴) نوشتن در ثبات‌های دستورالعمل

دستورات ۱، ۳، ۵، ۷، ۸، ۹، ۱۱ و ۱۲ در مد هسته اجرا شده و مابقی دستورات در مد کاربر اجرا می‌شوند.

۶- تفاوت وقفه (Interrupt) و تله (Trap) چیست؟

همان‌طور که در متن درس اشاره شد، وقفه‌ها به دو دسته خارجی و داخلی تقسیم می‌شوند که تله نوع خاصی از وقفه داخلی می‌باشد. معمولا تله (Trap) ناشی از یک خطا یا ایجاد یک شرایط استثنایی در اجرای فرایند جاری است.

۷- هدف مفسر فرمان چیست؟ چرا از هسته تفکیک می‌شود؟

مفسر فرمان، دستورات را از کاربر یا به عنوان مثال از یک فایل که شامل دستورالعمل‌های اجرایی است، گرفته و اجرا می‌کند. این عمل معمولا با یک یا بیش‌تر از یک فراخوان سیستمی آغاز شده و معمولا نیازی به اجرا در مد هسته ندارد و تنها زمانی که نیاز به اجرای دستورالعمل‌های ویژه باشد، پردازنده وارد مد هسته شده و در غیر این‌صورت بدون درگیر شدن هسته، اجرا می‌شوند.

۸- امتیاز رهیافت ریزهسته در طراحی سیستم‌عامل چیست؟

مزایای این روش عبارتند از:

الف- اضافه کردن یک سرویس جدید، نیازی به اصلاح هسته سیستم‌عامل ندارد، چرا که این عمل در خارج از هسته انجام می‌شود.

ب- طراحی هسته ساده‌تر و عملیاتی‌تر می‌شود.

ج- در صورت خرابی در سرویسی خارج از هسته، کل سیستم‌عامل از کار نمی‌افتد و فقط آن سرویس غیر فعال می‌شود. به همین دلیل، این ساختار به لحاظ میزان تحمل‌پذیری خطا (Fault tolerant)، بالا است.

د- مناسب برای سیستم‌های توزیع‌شده است.

۹- هدف از فراخوان‌های سیستمی (system calls) چیست و ارتباط آن‌ها با مفهوم عملکرد دو حالته (مد هسته و مد کاربر) چگونه است؟

در اصل فراخوان‌های سیستمی به برنامه‌های سطح کاربر اجازه می‌دهند که سرویس‌ها و توابعی را از سطح سیستم‌عامل درخواست کنند که این درخواست با تغییر مد از کاربر به هسته و سپس اجرای آن‌ها، انجام می‌شود.

۱۰- چه خصوصیتی در تله‌ها، وقفه‌ها، فراخوان‌های سیستمی و فراخوانی‌های زیر برنامه به صورت مشترک وجود دارد؟

تله‌ها، وقفه‌ها، فراخوان‌های سیستمی و فراخوانی‌های زیر برنامه، همگی مقدار جاری شمارنده برنامه (PC) را ذخیره کرده و به محل جدیدی از حافظه انشعاب می‌کنند.

۱۱- چه خصوصیتی در تله‌ها، وقفه‌ها و فراخوان‌های سیستمی به صورت مشترک وجود داشته و در فراخوانی‌های زیر برنامه وجود ندارد؟

تله‌ها، وقفه‌ها و فراخوان‌های سیستمی باعث می‌شوند که ماشین به مد هسته (Kernel mode) تغییر حالت دهد اما فراخوانی زیر برنامه، حالت اجرا را تغییر نمی‌دهد.

۱۲- چه خصوصیاتی در تله‌ها، فراخوان‌های سیستمی و فراخوانی‌های زیر برنامه مشترک هستند که در وقفه‌ها وجود ندارند؟ (تله را خارج از دسته‌بندی وقفه‌ها فرض کنید)

تله‌ها، فراخوان‌های سیستمی و فراخوانی‌های زیر برنامه، همگام هستند در حالی که وقفه‌ها ناهمگام‌اند.

۱۳- چه خصوصیاتی در فراخوان‌های سیستمی و فراخوانی‌های زیر برنامه مشترک هستند که در تله‌ها و وقفه‌ها وجود ندارند؟

فراخوان‌های سیستمی و فراخوانی‌های زیر برنامه، توسط برنامه‌های کاربر نمایان می‌شوند، ولی وقفه‌ها توسط سخت‌افزار نمایان می‌شوند و همچنین تله‌ها (همان‌طور که در متن درس اشاره شد، اصطلاح تله، حاکی از این است که برنامه خاطی توسط سخت‌افزار حفاظت در تله هسته گرفتار می‌شود).

۱۴- تقریبا در تمام سیستم‌هایی که مولفه‌های DMA دارند، اولویت دسترسی DMA به حافظه اصلی، بیش از اولویت پردازنده در دسترسی به حافظه است. چرا؟

زمانی که پردازنده مشغول خواندن یا نوشتن حافظه اصلی است، اگر دسترسی پردازنده از حافظه قطع شود معمولا اطلاعات کم و ناچیزی از بین می‌رود، زیرا پردازنده مدت زمان کمی را برای انتقال اطلاعات به‌طور محدود نیاز دارد. اما DMA، اطلاعات را در حجم بالا انتقال می‌دهد و قطع شدن دسترسی آن از حافظه، سبب از بین رفتن جریان قابل توجهی از اطلاعات خواهد شد. به همین دلیل، اولویت دسترسی DMA به حافظه اصلی، بیش‌تر از پردازنده است.

۱۵- یک مولفه DMA نویسه‌ها را با نرخ ۹۶۰۰ bps از یک دستگاه خارجی به حافظه اصلی منتقل می‌کند. پردازنده می‌تواند با نرخ یک میلیون دستورالعمل در ثانیه، دستورالعمل‌ها را واکشی کند. به‌خاطر فعالیت DMA پردازنده چقدر کند خواهد شد؟

اگر روال عملیات خواندن/نوشتن داده را کنار بگذاریم و فرض کنیم پردازنده تنها دستورالعمل‌ها را از حافظه اصلی واکشی می‌کند. از آن‌جا که طبق صورت مساله پردازنده می‌تواند با نرخ یک میلیون دستورالعمل در ثانیه دستورالعمل‌ها را واکشی کند، نتیجه می‌گیریم مدت زمان مورد نیاز برای واکشی یک دستورالعمل، یک میکروثانیه است و پردازنده هر یک میکروثانیه نیاز به دسترسی به حافظه اصلی دارد.

از آن‌جا که طبق صورت مسئله مولفه DMA نویسه‌ها را با نرخ ۹۶۰۰ bps از یک دستگاه خارجی به حافظه اصلی منتقل می‌کند، با فرض این‌که هر کاراکتر ۸ بیت است، پس نرخ انتقال اطلاعات ۱۲۰۰ کاراکتر بر ثانیه می‌باشد. به عبارتی دیگر، هر کاراکتر در مدت زمان ۸۳۳ میکروثانیه منتقل می‌شود. در واقع کنترل کننده‌ی DMA، از هر ۸۳۳ سیکل پردازنده، یک سیکل را به خود اختصاص می‌دهد.

پس پردازنده به طور تقریبی $\frac{۱}{۸۳۳} \times ۱۰۰\% = ۰/۱۲\%$ کند می‌شود.

تست‌های فصل اول

۱- فرض کنید زمان محاسبات یک فرایند ۲۰۰ سیکل cpu باشد. در ضمن عملیات I/O در حال انجام برای یک فرایند دیگر از طریق DMA در حال انجام بوده و پس از ۱۰۰ سیکل cpu، پایان عملیات I/O توسط یک وقفه به سیستم اطلاع داده شود. اگر زمان اجرای ISR را ۱۰ سیکل cpu فرض کنیم، کل عملیات مذکور چه مقدار از زمان سیستم را به خود اختصاص می‌دهند؟ (زمان هر سیکل cpu را معادل یک واحد زمانی فرض کنید.)

(IT - ۸۸)

۱) ۲۱۰ واحد زمانی
۲) کم‌تر از ۲۱۰ واحد زمانی
۳) ۳۱۰ واحد زمانی
۴) بیش‌تر از ۲۱۰ واحد زمانی

۲- کدام یک از دستورالعمل‌های زیر فقط قادر به اجرا در مد کرنل (Kernel Mode) است؟

(مهندسی کامپیوتر - ۸۶)

۱) خواندن ساعت سیستم
۲) خواندن PSW
۳) تنظیم زمان سیستم
۴) نوشتن در ثبات دستورالعمل

۳- کامپیوتری می‌تواند ۴ برنامه را برای اجرا به طور همزمان در حافظه داشته باشد. هر یک از این برنامه‌ها نیمی از وقت خود را منتظر عملیات ورودی و خروجی هستند. چه کسری از زمان پردازنده تلف می‌شود؟

(IT - ۸۶)

۱) $\frac{۱}{۱۶}$
۲) $\frac{۱}{۴}$
۳) $\frac{۱}{۲}$
۴) زمان تلف شده ندارد و پردازنده همواره مشغول اجرای یکی از ۴ برنامه است.

۴- کدام گزینه از وظایف سیستم‌عامل در رابطه با مدیریت سیستم I/O نیست؟

(IT آزاد - ۸۶)

۱) مدیریت اسپولینگ
۲) ایجاد و حذف فایل‌ها
۳) مدیریت بافرینگ
۴) مدیریت درایورها

۵- کدام گزینه از مزایای سیستم‌عامل ماشین مجازی نیست؟

(IT آزاد - ۸۶)

۱) سرعت بالا ۲) افزایش امنیت برای سیستم

۳) حل مشکلات ناسازگاری سیستم ۴) استفاده چندین کاربر از یک سیستم

۶- کدام‌یک از موارد زیر در یک سیستم Real Time درست است؟

(IT – ۸۵)

۱) تنها ملاک درستی انجام کار، آن است که در زمان مشخصی انجام شود.

۲) از حافظه مجازی به دلیل آن که زمان پردازش را طولانی می‌کند استفاده نمی‌شود.

۳) برای آن‌که بتوان به کارها با اولویت بالاتر پاسخ داد، پردازنده نمی‌تواند مدت زمان زیادی در Kernel باشد.

۴) اگر یک کار deadline نداشته باشد، ممکن است هیچ‌گاه پردازنده را در اختیار نگیرد، یعنی گرسنگی حاصل شود.

۷- کدام گزینه در مورد روش‌های I/O صحیح نیست؟

(IT – ۸۵)

۱) روش Programmed I/O پردازنده اصلی را درگیر عملیات I/O می‌کند.

۲) روش Interrupted I/O پردازنده اصلی را درگیر عملیات I/O می‌کند.

۳) روش DMA I/O پردازنده اصلی را درگیر عملیات I/O می‌کند (در حین I/O).

۴) روش I/O با استفاده از هر پردازنده خاص I/O، امکان نوشتن برنامه‌های مختلف از دستورات پردازنده اصلی و هم پردازنده I/O را می‌دهد.

۸- کدام گزینه در مورد ساختار سیستم‌عامل مبتنی بر مدل مشتری/ سرویس‌دهنده (Client/Server) صحیح نیست؟

(آزاد مهندسی کامپیوتر – ۸۳)

۱) این ساختار در مقابل رخداد خطا قابلیت اطمینان پایین دارد.

۲) این ساختار برای سیستم توزیع شده مناسب است.

۳) این ساختار بصورت لایه عمودی طراحی شده است.

۴) این ساختار مبتنی بر کم‌تر نمودن کدهای موجود در هسته (Kernel) است.

۹- کدام‌یک از عملیات زیر از تله (Trap) سیستم‌عامل استفاده نمی‌کند؟

(آزاد مهندسی کامپیوتر – ۸۲)

۱) فراخوانی سیستمی ۲) دسترسی غیر مجاز به حافظه

۳) مقداردهی متغیرها ۴) نقص صفحه

۱۰- کدام نوع سیستم‌عامل امکان ارایه ماشین مجازی محاوره‌ای را برای کاربران فراهم می‌کند؟

(آزاد مهندسی کامپیوتر - ۸۲)

۱) دسته‌ای ۲) کامپیوتر شخصی

۳) بلادرنگ ۴) اشتراک زمانی

۱۱- اگر Kernel شامل دو بخش وابسته به ماشین و مستقل از ماشین باشد، کدام گزینه دسته‌بندی مناسب‌تری از توابع Kernel است؟

(مهندسی کامپیوتر - ۸۱)

۱) گرداننده وقفه و زمانبندی فرایندها در بخش وابسته به ماشین و درایور (Driver) دستگاه‌ها و مدیریت فرایندها در بخش دیگر.

۲) گرداننده وقفه و درایور(Driver) دستگاه‌ها در بخش وابسته به ماشین و مدیریت و زمانبندی فرایندها در بخش مستقل از ماشین.

۳) مدیریت و زمانبندی فرایندها در بخش وابسته به ماشین و گرداننده وقفه و درایور دستگاه‌ها در بخش مستقل از ماشین.

۴) درایور دستگاه‌ها و زمانبندی فرایندها در بخش وابسته به ماشین و گرداننده وقفه و مدیریت فرایندها در بخش مستقل از ماشین.

۱۲- جدول زیر زمان‌های لازم برای ورود، محاسبه و خروج ۳ کار در یک سیستم دسته‌ای (Batch) با Spooling نشان می‌دهد. حداقل کل زمان مصرفی برای اجرای هر ۳ کار به شرط آنکه ورود کارها تعیین کننده ترتیب پردازش و ترتیب خروجی آن‌ها باشد چقدر است؟

(آزاد مهندسی کامپیوتر - ۷۹)

	زمان ورود	زمان پردازش	زمان خروج
کار اول	5	4	1
کار دوم	2	2	3
کار سوم	5	3	2

۱) ۲۷ ۲) ۱۰

۳) ۲۰ ۴) ۱۷

۱۳- از بخش‌های اصلی یک نرم‌افزار اداره‌کننده‌ی دستگاه (Device Drive)، می‌توان موارد زیر را برشمرد:

(مهندسی کامپیوتر - ۷۹)

۱) هسته سیستم‌عامل، زمانبند ورودی/خروجی، روتین‌های خدماتی وقفه‌های دستگاه

۲) مدیریت دستگاه، زمانبندی فرایندها و زمان‌بندی ورودی/خروجی

۳) تبدیل کننده استانداردهای ورودی/خروجی به یکدیگر، روتین‌های خدماتی وقفه، برنامه‌های ورودی/خروجی و هدایت دستگاه

۴) نرم‌افزارهای میانگیری (Buffering)، مدیریت پرونده‌ها، زمان‌بند ورودی/خروجی و حفاظت

۱۴-به یک سیستم اشتراک زمانی که امکان همزمانی حداکثر هشت پردازه را فراهم می‌کند، بیست عدد نوار گردان متصل است. تعداد زیاد و نامحدودی کار به سیستم ارجاع می‌شود که هر یک برای تکمیل عملیات خود به حداکثر چهار نوارگردان نیاز دارد. هر کار می‌تواند با در اختیار گرفتن سه نوارگردان شروع به کار کرده و برای مدت طولانی ادامه یابد، و در اواخر عملیات نوارگردان چهارم را درخواست کند. حال اگر زمان‌بند مورد استفاده در این سیستم اشتراک زمانی، تا زمانی که چهار گردانندهی نوار در دسترس نباشد شروع به کار نکند، و با شروع هر کار، بلافاصله چهار گرداننده به آن اختصاص داده و تا پایان عملیات آن کار آن‌ها را رها نکند، (الف) حداکثر تعداد کارهایی که می‌توانند به صورت همزمان در حال پیشرفت باشند، (ب) حداکثر نوارگردان‌هایی که بی‌کار می‌مانند و (ج) حداقل نوارگردان‌هایی که بی‌کار می‌مانند، چند تا است؟

(مهندسی کامپیوتر - ۷۸)

۱) (الف) هشت (ب) چهار (ج) یک ۲) (الف) شش (ب) دو (ج) دو

۳) (الف) هفت (ب) یک (ج) یک ۴) (الف) پنج (ب) پنج (ج) صفر

۱۵-در محیط یک سیستم‌عامل چند وظیفه‌ای (Multi-task) کدام یک از وقفه‌های زیر از اولویت بالاتری برخوردار می‌باشند؟

(آزاد مهندسی کامپیوتر - ۷۱)

۱) وقفه از طرف برنامه کاربر برای انجام I/O

۲) وقفه یک دستگاه جانبی برای اعلام پایان عمل I/O

۳) وقفه ساعت داخلی ماشین

۴) وقفه به خاطر سعی به دستیابی به آدرس غیرمجاز در حافظه اصلی

۱۶-کدام یک از تعاریف زیر مفهوم کامل‌تری از سیستم‌عامل را بیان می‌کند؟

(آزاد مهندسی کامپیوتر - ۷۱)

۱) سیستم‌عامل یک برنامه بزرگ کاربردی است که استفاده از سخت‌افزار را کامل می‌کند.

۲) سیستم‌عامل یک برنامه بزرگ سیستمی است که روی ماشین اجرا می‌شود.

۳) سیستم‌عامل یک برنامه سطح پایین است که از تک‌تک منابع سخت‌افزاری حداکثر استفاده را می‌برد.

۴) سیستم‌عامل یک برنامه است که مسئول استفاده کارآمد از کلیه منابع یک سیستم کامپیوتر می‌باشد.

۱۷- دو دلیل طراحی و پیاده‌سازی سیستم‌های عامل ماشین مجازی (Virtual Machine):

(آزاد مهندسی کامپیوتر - ۷۱)

۱) امکان برنامه‌نویسی به زبان‌های غیراسمبلی روی ماشین و شبیه‌سازی دستگاه‌های جانبی غیر موجود می‌باشد.

۲) اجرای همزمان چند سیستم‌عامل مختلف روی یک کامپیوتر و بالا بردن توان عملیاتی سیستم می‌باشد.

۳) تست یک سیستم‌عامل جدید همزمان با سیستم‌عامل قدیم یک کامپیوتر و شبیه‌سازی دستورات ماشینی که کامپیوتر فاقد آن می‌باشد.

۴) دو گزینه ۲ و ۳ صحیح است.

۱۸- در یک سیستم‌عامل گسترده (Distributed Operating System) کدام‌یک از موارد زیر درست نیست؟

(آزاد مهندسی کامپیوتر - ۷۱)

۱) چندین CPU مستقل از نظر جغرافیایی با هم فاصله دارند و تحت یک سیستم‌عامل کار می‌کنند.

۲) در تبادل پیغام کاربران می‌بایست آدرس ماشین‌های یکدیگر را بدانند.

۳) محل استقرار فایل‌ها در کنترل کاربران نمی‌باشد.

۴) قابلیت اطمینان یک سیستم‌عامل گسترده از یک سیستم‌عامل متمرکز بیش‌تر است.

۱۹- کدام‌یک از موارد زیر برای کامپیوترهای نسل دوم مصداق ندارد؟

(آزاد مهندسی کامپیوتر - ۷۱)

۱) زمان برگشت کار (Turnaround time) در سیستم بالا است.

۲) این کامپیوترها از لامپ خلا استفاده کرده و بسیار حجیم بودند.

۳) اغلب برنامه‌های کاربردی برای این سیستم‌ها به زبان فرترن نوشته می‌شد.

۴) گزینه ۱ و ۳ صحیح است.

۲۰- وظیفه یک سیستم‌عامل چند منظوره:

(آزاد مهندسی کامپیوتر - ۷۱)

۱) به اشتراک گذاشتن حافظه اصلی کامپیوتر می‌باشد.

۲) به اشتراک گذاشتن پردازنده و دستگاه‌های ورودی و خروجی است.

۳) حسابداری منابع مصرف شده در سیستم است.

۴) تمامی موارد

۲۱- کامپیوترهای آنالوگ کم‌ترین استفاده را در یکی از موارد زیر دارند:

(آزاد مهندسی کامپیوتر - ۷۱)

۱) کنترل پروسس‌های صنعتی ۲) کاربردهای پزشکی

۳) پردازش اطلاعات تجاری ۴) محاسبات فنی و مهندسی

۲۲- تکنیک Spooling به چه معناست:

(متفرقه)

۱) به‌کارگیری حافظه ثانویه به عنوان میانگیر حافظه هنگام پر شدن حافظه اصلی.

۲) به‌کارگیری حافظه ثانویه به عنوان میانگیر حافظه هنگام انتقال داده‌ها بین وسایل جانبی و پردازنده‌ها.

۳) به‌کارگیری حافظه ثانویه جهت ذخیره محاسبات پردازشگرها هنگام پر شدن حافظه اصلی.

۴) به کارگیری حافظه اصلی به عنوان یک میانگیر حافظه ثانویه جهت کاهش تاخیرهای پردازش.

۲۳- در کدام یک از سیستم‌های عامل زیر، برنامه به چند پردازش مجزا تقسیم می‌شود؟

(متفرقه)

۱) Multi tasking ۲) Multi processing.

۳) Multi programming ۴) Time sharing

۲۴- کدام یک از موارد زیر جز سیستم‌های Multi Acess نیست؟

(متفرقه)

۱) Batch processing ۲) Real time

۳) Time sharing ۴) Transactin processing

۲۵- یک سیستم‌عامل چند کاربره کدام یک از قابلیت‌های زیر را دارد؟

(متفرقه)

۱) اجازه می‌دهد که چند کاربر به صورت شبکه در حالت غیر همزمان از آن استفاده کنند.

۲) اجازه می‌دهد که چند کاربر به صورت غیر همزمان از کامپیوتر استفاده کنند.

۳) اجازه می‌دهد که چند کاربر به صورت همزمان از کامپیوتر استفاده کنند.

۴) تمامی موارد فوق

۲۶- کدام یک از موارد زیر از مزایای Offline spooling می‌باشد؟

(متفرقه)

۱) عدم نیاز به سخت‌افزار اضافی.

۲) زمان گردش کار کوتاه.

۳) سهولت برای استفاده از راه دور.

۴) ایجاد اولویتی در دسترسی به کامپیوتر.

۲۷- مزیت Online spooling نسبت به Offline spooling در چیست؟

(متفرقه)

۱) عملیان آن ساده‌تر است.

۲) دسترسی با اولویت امکان‌پذیر است.

۳) ارزان‌تر است.

۴) در استفاده از راه دور سهولت آن بیشتر است.

۲۸- در سیستم‌عامل دستورالعمل‌های زبان اسمبلی که واسطه‌ی ما بین فرایند و سیستم‌عامل را فراهم می‌سازند، چه نامیده می‌شوند؟

(متفرقه)

۱) System calls ۲) Shell

۳) Kernel ۴) Calls

۲۹- کدام گزینه از مزایای ساختار سیستم‌عامل لایه‌ای نسبت به ساختار سیستم‌عامل یکپارچه نیست؟

(متفرقه)

۱) قابلیت گسترش بیش‌تر ۲) سرعت بیش‌تر

۳) خطایابی ساده‌تر ۴) مدیریت ساده‌تر

۳۰- ضروری‌ترین هدف سیستم‌عامل چیست؟

(متفرقه)

۱) مدیریت منابع

۲) ایجاد سهولت کار برای کاربران

۳) سهولت گسترش در مقابل سخت‌افزار جدید

۴) سهولت مقابله با خطاهای جدید

۳۱- کدام گزینه در مورد بخش ورودی و خروجی (I/O) درست نیست؟

(متفرقه)

۱) وقفه‌ها در پایین‌ترین سطح مدیریت I/O و نزدیک به سخت‌افزار هستند.

۲) Device driverها (گرداننده‌های دستگاه) وابسته به سخت‌افزار آن دستگاه I/O هستند.

۳) در فضای سیستم‌عامل، بخشی از نرم‌افزار I/O مستقل از سخت‌افزار وجود دارد.

۴) در فضای کاربر و خارج از فضای سیستم‌عامل، نرم‌افزار I/O وجود ندارد.

۳۲- کدام گزینه در مورد سیستم‌عامل اشتراک زمانی در Mainframe، درست نیست؟

(متفرقه)

۱) قابلیت چند کاربره (Multi user) را فراهم می‌کند.

۲) قابلیت چند برنامگی (Multi program) را فراهم می‌کند.

۳) ارتباط کاربر به صورت Offline است.

۴) به صورت محاوره‌ای (Interactive) است.

۳۳- در کدام ساختار سیستم‌عامل، قسمتی از سرویس‌های سیستم‌عامل به جای این‌که در فضای هسته قرار داشته باشند در فضای کاربر قرار می‌گیرد؟

(متفرقه)

۱) لایه‌ای
۲) Client/Server
۳) یکپارچه
۴) ماشین مجازی

۳۴- کدام گزینه درست نیست؟

(متفرقه)

۱) پوسته (Shell) وظیفه ارتباط سیستم‌عامل با منابع سخت‌افزار را برعهده دارد.

۲) پوسته وظیفه ارتباط سیستم‌عامل با کاربران را برعهده دارد.

۳) هسته (Kernel) وظیفه ارتباط سیستم‌عامل با منابع سخت‌افزار را برعهده دارد.

۴) هسته وظیفه ارتباط سیستم‌عامل با منابع نرم‌افزاری را برعهده دارد.

۳۵- کدام گزینه در مورد فراخوانی سیستم (System call) درست نیست؟

(متفرقه)

۱) ارتباط بین برنامه‌های کاربردی و سیستم‌عامل را فراهم می‌کند.

۲) برای دسترسی به منابع نرم‌افزاری استفاده می‌شوند.

۳) روتین‌های نوشته شده کاربر برای ارتباط با برنامه‌های کاربردی است.

۴) برای دسترسی به منابع سخت‌افزاری استفاده می‌شوند.

۳۶- در سیستم‌های اشتراک زمانی چرا کاربران می‌توانند با برنامه در حال اجرا محاوره و تعامل داشته باشند؟

(متفرقه)

۱) زیرا زمان اختصاص یافته به هر فرایند طولانی است.

۲) زیرا زمان اجرای فرایندها قابل پیش‌بینی است.

۳) زیرا فرایندهایی با زمان اجرای کم‌تر زودتر اجرا می‌شوند.

۴) زیرا سوییچ کردن بین فرایندها بسیار سریع اتفاق می‌افتد.

۳۷- در یک سیستم اشتراک زمانی، منبع دستورات به سیستم‌عامل چیست؟

(متفرقه)

۱) دستورالعمل‌های JCL که همراه برنامه ارائه می‌شود.

۲) دستورالعمل‌هایی خاص که همراه برنامه ارائه می‌شود.

۳) فرمان‌هایی که از پایانه وارد می‌شود.

۴) هیچ کدام.

۳۸- کدام گزینه نادرست است؟

(متفرقه)

۱) سیستم‌عامل برنامه کنترلی می‌باشد که اجرای برنامه‌های کاربر و سیستم را کنترل می‌کند تا از خطاها و استفاده غیر صحیح از کامپیوتر جلوگیری نماید.

۲) سیستم‌عامل برنامه‌ای سیستمی است که در اکثر اوقات بر روی کامپیوتر در حال اجرا است.

۳) از اهداف سیستم‌عامل مدیریت منابع و سهولت استفاده از سیستم کامپیوتری است.

۴) هیچ کدام.

۳۹- در سیستم‌عامل اشتراک زمانی (Time sharing) وقت پردازنده در چه صورت بین برنامه‌ها تقسیم می‌شود؟

(متفرقه)

۱) با توجه به اولویت برنامه.

۲) تخصیص وقت پردازنده به طور تصادفی انجام می‌گیرد.

۳) در این سیستم‌ها وقت پردازنده به طور مساوی و در مقاطع زمانی مشخص و محدود بین برنامه‌ها تقسیم می‌شود.

۴) هیچ‌کدام.

۴۰- کدام یک از انواع وقفه‌ها به سیستم‌عامل امکان می‌دهد بعضی از اعمال نظیر چک کردن سخت‌افزار را به شکل مرتب و در یک پریود زمانی خاص انجام دهد؟

(متفرقه)

۱) I/O ۲) Timer

۳) Machine check ۴) Program check

۴۱- مزیت استفاده از Spooling کدام است؟

(متفرقه)

۱) کاهش پیچیدگی سیستم‌عامل ۲) صرفه‌جویی در حافظه

۳) همزمانی انجام عمل I/O و پردازش ۴) هیچ‌کدام

۴۲- کدام گزینه درباره سیستم‌های اشتراک زمانی صحیح نیست؟

(متفرقه)

۱) اجرای موازی (Parallel) قسمت محاسباتی دو فرایند امکان‌پذیر است.

۲) اجرای موازی قسمت محاسباتی یک فرایند با عمل I/O فرایند دیگر امکان‌پذیر است.

۳) اجرای همروند (Concurrent) قسمت محاسباتی دو فرایند امکان‌پذیر است.

۴) اجرای همروند عمل I/O دو فرایند امکان‌پذیر است.

۴۳- کدام گزینه در مورد سیستم‌های غیر قطعی نادرست است؟

(متفرقه)

۱) عملکرد استفاده متغیر و گوناگون است.

۲) این نوع سیستم‌ها معمول‌تر و رایج‌تر هستند.

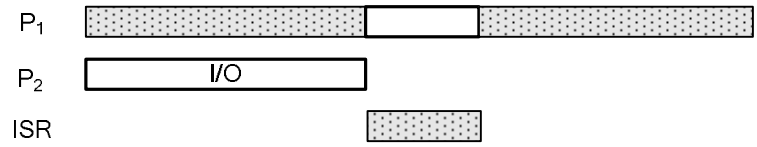
۳) شامل سیستم‌های همه منظوره‌ی اشتراک زمانی می‌شوند.

۴) تعداد فعل و انفعالات و زمان پردازنده‌ی مورد نیاز هر کار از قبل معلوم است.

پاسخ تست‌های فصل اول

۱- گزینه ۴ صحیح است.

فرض کنید فرایندی به نام P_1، به اندازه ۲۰۰ سیکل cpu، پردازنده را جهت محاسباتش نیاز داشته باشد و فرایند P_2 نیز به اندازه‌ای معادل با ۱۰۰ سیکل cpu، مشغول انجام عملیات I/O باشد. چون از DMA استفاده شده است، عمل I/O فرایند P_2 می‌تواند همزمان با عمل محاسبات P_1 انجام شود، که نحوه‌ی اجرای این دو فرایند و همچنین پردازش ISR مربوطه، مشابه شکل زیر خواهد شد:



همان‌طور که در شکل مشخص است، ۲۰۰ سیکل cpu محاسبات فرایند P_1 و ۱۰ سیکل cpu، صرف اجرای ISR مربوط به فرایند P_2 خواهد شد که در مجموع ۲۱۰ سیکل cpu می‌شود. اما از آن‌جا که در هنگام کار با DMA و انجام عمل I/O، در زمان‌هایی گذرگاه کاملا در اختیار DMA قرار می‌گیرد، می‌تواند زمان اجرا شدن فرایند P_1 بیش‌تر از ۲۰۰ سیکل cpu شود. لذا کل عملیات این دو فرایند، بیش‌تر از ۲۱۰ سیکل cpu خواهد شد.

۲- گزینه ۳ صحیح است.

تنظیم زمان سیستم، نیاز به اجرا در مود کرنل دارد. برای بررسی دقیق‌تر به مسائل فصل ۱ مراجعه شود.

۳- گزینه ۱ صحیح است.

اگر زمان هدر رفتن پردازنده برای هر برنامه $\frac{۱}{۲}$ در نظر گرفته شود، زمان کل هدر رفته برابر خواهد بود با:

$$P = \frac{۱}{۲} \times \frac{۱}{۲} \times \frac{۱}{۲} \times \frac{۱}{۲} = \frac{۱}{۱۶}$$

همچنین بهره‌وری پردازنده در این سیستم به صورت زیر محاسبه می‌شود:

$$1-P = 1-\frac{۱}{۱۶} = \frac{۱۵}{۱۶}$$

۴- گزینه ۲ صحیح است.

۵- گزینه ۱ صحیح است.

۶- گزینه ۴ صحیح است.

درست انجام شدن یک کار به عوامل مختلفی بستگی دارد که یکی از آن‌ها در سیستم‌های بلادرنگ، در زمان مشخص انجام شدن است. همچنین در این سیستم‌ها بهتر است از حافظه مجازی استفاده نشود، چون زمان پردازش را طولانی می‌کند. لذا سه گزینه اول غلط بوده و تنها گزینه درست، ۴ است.

۷- گزینه ۳ صحیح است.

با استفاده از DMA در حین عملیات ورودی و خروجی، پردازنده درگیر عملیات نیست و می‌تواند به اجـرای سـایر برنامه‌ها بپردازد.

۸- گزینه ۱ صحیح است.

در این ساختار بخش‌های اصلی از غیر اصلی جدا شده و درون هسته قرار می‌گیرنـد، لـذا قابلیـت اطمینـان آن در مقابل بروز خطا بالا است. پس گزینه‌ی ۱ نادرست است.

می‌دانیم که در طراحی سیستم‌های توزیع شده، می‌توان وظایف مختلف و مستقل را برای اجرا بین چنـد کـامپیوتر توزیع کرد. به این ساختار، طراحی به صورت لایه عمودی گفته می‌شود و لذا گزینه ۳ درست اسـت. درسـت بـودن گزینه‌های ۲ و ۴، بدیهی است.

۹- گزینه ۳ صحیح است.

تله سیستم‌عامل، وقفه‌ای است که هنگام اجرای برنامه‌ها (در مود کاربر) به وجود آمده و باعث توقـف پردازنـده از اجرای برنامه جاری و تخصیص آن به سیستم‌عامل می‌شود. در ادامه، سیستم‌عامل با بررسی تله، سـرویس مربوطـه و مورد نیاز را اجرا می‌کند (اجرای تله مورد نظر).

بنابراین با توجه به گزینه‌های داده شده، پر واضح است که، فراخوان سیستمی، دسترسی غیـر مجـاز بـه حافظـه و نقص صفحه (نقص صفحه در فصل ۵ توضـیح داده خواهـد شـد) بـه تلـه سیسـتم‌عامـل نیـاز دارنـد و در مقابـل مقداردهی متغیرها هیچ‌گونه ارتباطی با تله سیستم‌عامل ندارد.

۱۰- گزینه ۴ صحیح است.

در سیستم‌عامل‌های اشتراک زمانی، می‌توان پردازنده یک کامپیوتر را (و همچنین سایر منابع) به طـور مجـازی بـین تعدادی کاربر به اشتراک گذاشت به گونه‌ای که هر کاربر بتواند به صورت محاوره‌ای از آن استفاده کند.

۱۱- گزینه ۲ صحیح است.

۱۲- گزینه صحیح وجود ندارد.

از آن‌جایی‌که در سیستم‌های Spooling امکان پردازش یک کار با ورودی و خروجـی چنـد کـار دیگـر بـه‌طـور همزمان امکان‌پذیر است، تست را حل می‌کنیم:

با توجه به زمان ورودی داده شده، ابتدا کار ۲ وارد شده و ۲ واحد زمانی اجرایش طول می‌کشد، یعنی تـا زمـان ۴. زمان ورود دو کار دیگر، ۵ است که در این فاصله ۱ واحد زمانی از خروجی کار اول انجام می‌شـود و مـابقی آن بـا اجرای کار بعدی انجام می‌گیرد. در زمان ۵ امکان ورود کار ۱ و کار ۳ وجود دارد که با توجه بـه تاکیـد تسـت مبنـی بر حداقل نمودن زمان کل مصرفی، ابتدا کار ۳ اجرا می‌شود و پس از طی ۳ واحد زمانی اجرایش تمام خواهـد شـد (خروجی این کار نیز با انجام کار بعدی به طور همزمان انجام می‌گیرد). در انتها کار ۱ اجرا شده کـه ۴ واحـد زمـانی طول می‌کشد و ۱ واحد زمانی هم برای خروجی آن مورد نیاز است. پس مجموع زمان‌های مصرفی عبارت‌اند از:

کل زمان مصرفی = ۲+۱+۳+۴+۱=۱۱

که در گزینه‌ها موجود نمی‌باشد.

۱۳- گزینه ۳ صحیح است.

زمان‌بندی ورودی و خروجی، برعهده بخش زمان‌بندِ I/O در سیستم‌عامل است. هسته سیستم‌عامل و زمان‌بندی فرایندها از بخش‌های نرم‌افزار اداره‌کننده‌ی دستگاه نیستند.

۱۴- گزینه ۴ صحیح است.

در ابتدا سیستم با چهار نوارگردان آزاد شروع به کار می‌کند، در نتیجه ابتدا حداکثر به طور همزمان ۵ کار را اجرا می‌کند. از آن‌جا که کارها مدت زمان زیادی نوارگردان چهارم را درخواست نمی‌کنند، در نتیجه حداکثر ۵ نوارگردان در سیستم می‌توانند بیکار باشند و زمانی که نوارگردان‌ها در اختیار کارها قرار گرفته باشند، مقدار نوارگردان‌های آزاد صفر خواهد بود.

۱۵- گزینه ۳ صحیح است.

۱۶- گزینه ۴ صحیح است.

سیستم‌عامل برنامه کاربردی نیست، بلکه یک برنامه سیستمی است، لذا گزینه ۱ صحیح نیست. دو تعریف گفته شده در گزینه‌های ۲ و ۳ می‌توانند صحیح باشند، اما از آن‌جا که در صورت تست کامل‌ترین مفهوم سیستم‌عامل خواسته شده است، صحیح‌ترین گزینه، گزینه ۴ می‌باشد.

۱۷- گزینه ۴ صحیح است.

جمله‌ی "شبیه‌سازی دستگاه‌های جانبی غیر موجود" که در گزینه ۱ آمده است صحیح نمی‌باشد. همان‌طور که در متن درس اشاره شده است، دلائل ذکر شده در دو گزینه ۲ و ۳ می‌توانند صحیح باشد.

۱۸- گزینه ۲ صحیح است.

همان‌طور که در متن درس اشاره شد، یکی از خواص سیستم‌عامل‌های توزیع شده، این است که سیستم باید از دیدگاه کاربر شفاف (یا نامریی) باشد. به این معنا که بتوان هر چیز را با نام آن فراخوانی کرد و نیازی به آدرس آن نباشد.

۱۹- گزینه ۲ صحیح است.

در این نسل (نسل دوم/سیستم‌های ساده دسته‌ای) از ترانزیستور به جای لامپ خلا استفاده می‌شد و اغلب برنامه‌ها به زبان فرترن یا اسمبلی نوشته می‌شدند و همچنین زمان برگشت کار در این سیستم‌ها بالا بود.

۲۰- گزینه ۴ صحیح است.

۲۱- گزینه ۳ صحیح است.

کامپیوترهای آنالوگ، کامپیوترهایی هستند که از کمیت‌های پیوسته ولتاژ و جریان استفاده می‌کنند. لذا می‌توان تا حدودی از آن‌ها در کارهای صنعتی، پزشکی (مانند دستگاه فشار خون) و محاسباتی استفاده کرد. اما استفاده از این نوع کامپیوترها در پردازش اطلاعات تجاری به ندرت انجام می‌شود.

۲۲- گزینه ۲ صحیح است.

۲۳- گزینه ۱ صحیح است.

۲۴- گزینه ۱ صحیح است.

۲۵- گزینه ۳ صحیح است.

۲۶- گزینه ۳ صحیح است.

همان‌طور که در متن درس اشاره شد، در روش Offline spooling کاربر می‌تواند برنامه خود را برای اجرا در کامپیوتر، به یک اپراتور سپرده و پس از مدتی به دنبال نتایج آن بیاید. پس این روش قابلیت استفاده از راه دور را دارد.

۲۷- گزینه ۲ صحیح است.

۲۸- گزینه ۱ صحیح است.

۲۹- گزینه ۲ صحیح است.

لایه‌ای بودن سیستم‌عامل، سبب کاهش سرعت عملیات می‌شود.

۳۰- گزینه ۱ صحیح است.

مهم‌ترین هدف یک سیستم‌عامل، مدیریت منابع فیزیکی و منطقی است.

۳۱- گزینه ۴ صحیح است.

۳۲- گزینه ۳ صحیح است.

۳۳- گزینه ۲ صحیح است.

۳۴- گزینه ۱ صحیح است.

۳۵- گزینه ۳ صحیح است.

۳۶- گزینه ۴ صحیح است.

۳۷- گزینه ۳ صحیح است.

از آن‌جا که در یک سیستم اشتراک زمانی، کاربران از طریق پایانه‌ها با سیستم در ارتباط هستند، بنابراین منبع دستورات به سیستم‌عامل، فرمان‌هایی هستند که از طریق پایانه وارد می‌شوند.

۳۸- گزینه ۴ صحیح است.

۳۹- گزینه ۳ صحیح است.

۴۰- گزینه ۲ صحیح است.

۴۱- گزینه ۳ صحیح است.

۴۲- گزینه ۱ صحیح است.

در سیستم‌های اشتراک زمانی، زمان یک پردازنده بین فرایندها تقسیم می‌شود. پس اجرای موازی فرایندها امکان‌پذیر نیست.

۴۳- گزینه ۴ صحیح است.

سیستم‌های غیر قطعی، سیستم‌هایی هستند که تعداد فعل و انفعالات و زمان پردازنده مورد نیاز هر کار از قبل معلوم باشد.

سؤال‌های مهندسی کامپیوتر-۱۳۹۹

۱- **کدام گزینه درباره ریسه‌های (Threads) سطح کاربر و سطح هسته درست است؟**

۱) زمان‌بندی ریسه‌های سطح هسته سریعتر از ریسه‌های سطح کاربر است.

۲) ریسه‌های سطح کاربر و سطح هسته از طریق فراخوانی سیستمی (System calls) به هم سرویس می‌دهند.

۳) ریسه‌های سطح کاربر و سطح هسته می‌توانند به فضای آدرس هم دسترسی داشته و می‌توانند در فضای آدرس هم بنویسید.

۴) ریسه‌های سطح هسته به فضای آدرس ریسه‌های سطح کاربر دسترسی دارند، اما ریسه‌های سطح کاربر به فضای آدرس ریسه‌های سطح هسته دسترسی ندارند.

۲- **در یک سیستم عامل از صفحه‌بندی وارون (Inverted paging) استفاده می‌شود. کدام گزینه در مورد جدول صفحه درست است؟**

۱) یک جدول صفحهٔ عمومی که براساس شماره قاب مرتب شده است.

۲) یک جدول صفحهٔ عمومی که براساس شماره پردازه مرتب شده است.

۳) یک جدول صفحهٔ عمومی که براساس شماره آدرس مجازی مرتب شده است.

۴) هر پردازه دارای یک جدول صفحهٔ اختصاصی است که براساس شماره قاب مرتب شده است.

۳- **در یک سیستم که تخصیص حافظه در آن براساس صفحه‌بندی (Paging) انجام می‌شود، اندازهٔ هر فریم (Frame) برابر ۲۰۴۸ بایت است. شکل زیر، حافظهٔ اصلی سیستم است، که قسمت‌های خاکستری فریم‌های تخصیص داده شده به یک پردازه هستند. اگر Internal Fragmentation برابر ۹۰۰ بایت باشد، اندازهٔ پردازه و External Fragementation چند بایت است؟**

۱) اندازهٔ پردازه برابر ۱۲۲۸۸ بایت و اندازهٔ External Fragmentation برابر ۱۴۳۳۶ بایت است.

۲) اندازهٔ پردازه برابر ۱۲۲۸۸ بایت و اندازهٔ External Fragmentation برابر صفر است.

۳) اندازهٔ پردازه برابر ۱۱۳۸۸ بایت و اندازهٔ External Fragmentation برابر صفر است.

۴) اندازهٔ پردازه برابر ۶ بایت و اندازهٔ External Fragmentation برابر ۷ بایت است.

۴- **در یک سیستم صفحه‌بندی (paging)، طول آدرس منطقی ۱۹ بیت است. اگر تعداد صفحات موجود در فضای آدرس منطقی ۱۲۹ باشد و قرار باشد به یک فضای آدرس فیزیکی ۱ مگابایتی نگاشت صورت گیرد،**

هر مدخل (entry) از جدول صفحه (Page table) باید چند بیت باشد؟ (بدون در نظر گرفتن بیت valid یا invalid در page table)

۱) ۸ ۲) ۹ ۳) ۱۷ ۴) ۲۰

۵- در یک سیستم زمان‌بندی MLFQ دو صف RR با برش زمانی ۳ و ∞ وجود دارد. میانگین زمان بازگشت برای پردازش‌های زیر کدام است؟

	arrival time	CPU Burst ۱	I / O Burst ۱	CPU Burst ۲
p۱	۰	۳	۶	۲
p۲	۳	۷	۵	۱
p۳	۴	۲	۴	۴
p۴	۱۱	۵	۳	۱

۱) ۲۱/۵ ۲) ۱۷ ۳) ۱۶/۵ ۴) ۱۴

۶- یک سامانه مدیریت حافظه را در نظر بگیرید، که تنها چهار قاب به پردازه اختصاص داده شده است که در ابتدا خالی هستند. حال پردازه صفحات را به ترتیب از چپ به راست ۷، ۶، ۱، ۲، ۲، ۳، ۴، ۳، ۲، ۱ ارجاع می‌کند. اگر این سیستم مدیریت حافظه از روش LRU برای جایگزینی صفحه استفاده کند، چه تعداد نقص صفحه خواهیم داشت؟

۱) ۷ ۲) ۸ ۳) ۹ ۴) ۱۰

۷- در یک دستگاه دیسک‌خوان، بازو روی سیلندر ۳۵ قرار دارد و جهت حرکت آن به سمت شیارهای بزرگتر با شماره بزرگتر است. اگر زمان انتقال از یک سیلندر به بعدی ۵ms باشد و از الگوریتم آسانسور برای دستیابی به سیلندرها استفاده شود، کل زمان جستجو برای دستیابی به سیلندرهای زیر (به ترتیب از چپ به راست) چند میلی‌ثانیه است؟

۴۴ و ۲۵ و ۱۲ و ۸۰ و ۳۹ و ۴۵ و ۳۴

۱) ۵۶۵ ۲) ۵۷۵ ۳) ۶۶۵ ۴) ۶۷۵

۸- کدام عبارت ذیل نمی‌تواند خروجی اجرای هم‌روند فرایندهای P_1 و P_2 باشد؟ (s۱ متغیر سراسری با مقدار اولیه صفر می‌باشد.)

P_1	P_2
Print "A"	Print "C"
wait (s۱)	signal (s۱)
Print "B"	Print "D"

۱) ABCD
۲) ACDB
۳) CABD
۴) CADB

پاسخنامه سؤال‌های مهندسی کامپیوتر-۱۳۹۹

۱- گزینه «۳» صحیح است.

زمان‌بندی نخ کاربر از هسته سریعتر است (گزینه ۱ غلط). نخ‌های کاربر و هسته از طریق lwp به هم نگاشت می‌شوند و سیستم کال برای این منظور نیست (گزینه ۲ غلط). نخ‌های کاربر و هسته می‌توانند به فضای آدرس هم دسترسی داشته باشند و در فضای آدرس هم بنویسند.

۲- گزینه «۱» صحیح است.

جدول صفحه وارون براساس شماره قاب و نه شمارهٔ صفحه مرتب شده است در ضمن جدول صفحه وارون فقط یک جدول است برای همهٔ فرآیندها و نه اینکه هر فرآیند جدول صفحه وارون داشته باشد.

۳- گزینه «۳» صحیح است.

تکه تکه شدن خارجی در صفحه‌بندی وجود ندارد و صفر بایت است. طبق شکل، ۶ قاب به فرآیند تخصیص داده شده و هر قاب ۲۰۴۸ بایت است ولی چون سوال گفته ۹۰۰ بایت تکه تکه شدن داخلی داریم یعنی از آخرین قاب مربوط به فرآیند ۹۰۰ بایت خالی است پس اندازه فرآیند برابر است با:

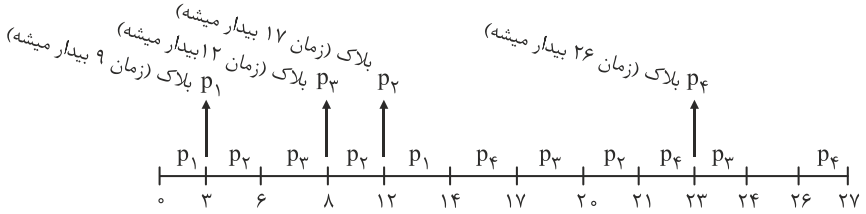
$$۶\times۲۰۴۸-۹۰۰=۱۱۳۸۸\ \text{byte}$$

۴- گزینه «۲» صحیح است.

آدرس منطقی ۱۹ بیت است و چون ۱۲۹ صفحه وجود دارد پس فیلد شماره صفحه ۸ بیتی است پس ۱۱ بیت آفست است. فضای آدرس فیزیکی ۱Mbyte است پس آدرس فیزیکی ۲۰ بیت است و چون ۱۱ بیت آفست است پس ۹ بیت شمارهٔ قاب است بنابراین هر مدخل در جدول صفحه که شامل شمارهٔ قاب است ۹ بیت است. توجه کنید که بیت‌های کنترلی نادیده گرفته شده‌اند.

۵- گزینه «۲» صحیح است.

در روش MLFQ هر فرآیند ابتدا به صف اول وارد می‌شود که در این جا RR با برش ۳ است. اگر برش فرآیند تمام شود به صف دوم که FFFO است می‌رود اگر فرایند در هر صفی باشد و بلاک شود (بخاطر I/O مثلاً) پس از بیداری به صف اول می‌رود. چارت گانت به شکل زیر است:



$$\text{ATT}=\frac{(۱۴-۰)+(۲۱-۳)+(۲۴-۴)+(۲۷-۱۱)}{۴}=۱۷$$

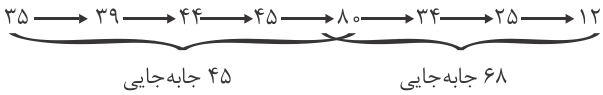
۶- این تست حذف شد.

۱	۲	۳	۴	۳	۲	۲	۱	۶	۷
×	×	×	×	✓	✓	✓	✓	×	×

تعداد ۶ تا نقص صفحه دارد که در گزینه‌ها نیست.

۷- گزینه «۱» صحیح است.

آسانسور همان look است (شماره آخرین سیلندر داده نشده است)



تعداد کل جابجایی $= ۴۵ + ۶۸ = ۱۱۳$

زمان کل $= ۱۱۳ \times ۵ = ۵۶۵ \text{ ms}$

۸- گزینه «۱» صحیح است.

اگر از p_1 شروع کنیم ابتدا A چاپ می‌شود ولی بعد، p_1 روی s_1 می‌خوابد (s_1 صفر است) پس بعد از A قطعاً B چاپ نمی‌شود. حال نوبت p_2 که شود C را چاپ می‌کند و سپس با سیگنال p_1 بیدار شده و به صف آماده می‌رود که از این به بعد D بعد B یا برعکس می‌توانند چاپ شوند یعنی خروجی از چپ به راست ACDB و ACBD امکان‌پذیر است حال اگر p_2 ابتدا اجرا شود C را چاپ می‌کند حال بعد از آن A و D و B یا D و A و B یا A و B و D امکان چاپ دارند.

سؤال‌های IT -۱۳۹۹

۱- **کدام گزینه از مزایای ساختار سیستم عامل لایه‌ای (Layered) نسبت به ساختار سیستم عامل یکپارچه (Monolithie) نیست؟**

۱) قابلیت گسترش بیشتر
۲) خطایابی ساده‌تر
۳) مدیریت ساده‌تر
۴) سرعت بیشتر

۲- **کدام مورد از مزایای ساختار ریزهسته (Miero Kernel) در طراحی سیستم عامل نیست؟**

۱) کارایی سیستم را افزایش می‌دهد.
۲) برای سیستم‌های توزیع شده مناسب است.
۳) اضافه کردن سرویس جدید نیازی به اصلاح هستهٔ سیستم عامل ندارد.
۴) در صورت بروز خرابی در سرویس خارج از هسته، کل سیستم عامل از کار نمی‌افتد.

۳- **با توجه به جدول زیر، متوسط زمان پاسخ‌دهی (Response Time) و متوسط زمان انتظار (Waiting Time) پردازه‌ها برای الگوریتم Preemptive Shortest Remaining Job First چند واحد زمانی است؟**

پردازه	زمان محاسبات	زمان ورود پردازه
P_1	۵	۳
P_2	۸	۱
P_3	۶	۲

۱) متوسط زمان پاسخ‌دهی برابر ۶ و متوسط زمان انتظار برابر صفر است.

۲) متوسط زمان پاسخ‌دهی برابر ۶ و متوسط زمان انتظار برابر ۱/۶ است.

۳) متوسط زمان پاسخ‌دهی برابر ۶ و متوسط زمان انتظار برابر ۶ است.

۴) متوسط زمان پاسخ‌دهی برابر ۶/۶ و متوسط زمان انتظار برابر ۶ است.

۴- **در یک سیستم، ۵۰ پردازه (Process) با کد زیر به صورت هم‌روند (Coneurrent) در حال اجرا هستند. اگر مقدار اولیه سمافورها $y = ۱۵$، $x = ۲۰$ و $z = ۱$ باشند، حداکثر چند پردازه ممکن است پشت سمافور z در حالت انتظار بلوکه شوند؟**

```
Wait (x);
Wait (y);
Wait (z);
a = a + ۱;
signal (z);
signal (y);
signal (x);
```

۱) ۱۴

۲) ۳۰

۳) ۴۹

۴) ۵۰

۵- **یک سامانه دارای ۶۴ صفحهٔ مجازی (Virtual pages) است که به ۱۶ قاب فیزیکی (Physical frames) براساس رابطهٔ زیر نگاشت داده می‌شود. طول هر صفحه یک کیلو کلمه (۱k Words) است. اگر آدرس مجازی برابر ۱۰۱۰۱۰۱۰۰۰۱۱۱۱۰۱ باشد، کدام گزینه آدرس فیزیکی را نشان می‌دهد؟**

۱) ۱۰۱۰۱۰۱۰۰۰۱۱۱۱

۲) ۱۰۱۱۱۰۰۰۱۱۱۱۰۱

۳) ۱۰۱۰۱۰۰۰۱۱۱۱۰۱

۴) ۱۱۰۰۱۰۰۰۱۱۱۱۰۱

۶- **کدام گزینه معیار (Criterion) یک زمان‌بند پردازنده نیست؟**

۱) زمان پاسخ

۲) بهره‌وری پردازنده

۳) گذردهی (Throughput)

۴) زمان Brust (Brust time)

۷- **برای خواندن از دیسک، در کدام لایه نرم‌افزاری محاسبات مربوط به شیار (Track)، قطاع (Sector) و هد دیسک صورت می‌پذیرد؟**

۱) لایه Device Driver

۲) لایه روتین سرویس‌دهی به وقفه

۳) لایه مدیریت دستگاه‌های سیستم عامل

۴) لایه نخ سطح هسته که برنامه سطح کاربر را اجرا می‌کند.

پاسخنامه سؤال‌های IT –۱۳۹۹

۱- گزینه «۴» صحیح است.

هرچه تعداد لایه‌ها بیشتر شود سرعت کمتر و کارایی کمتر می‌شود.

۲- گزینه «۱» صحیح است.

کارایی و سرعت پایین است چون اکثر درخواست‌های کاربر نیاز به send و receive دارد.

۳- تست حذف شد.

متوسط زمان محاسبات $\text{ABT} = \frac{۵+۸+۶}{۳} = ۶/۳$ است که جمع آن با متوسط زمان انتظار باید برابر متوسط زمان پاسخ‌دهی شود که هیچ گزینه‌ای چنین نیست.

۴- گزینه «۱» صحیح است.

از سمافور x تعداد ۲۰ فرآیند عبور می‌کنند از این تعداد ۱۵ فرآیند از y عبور می‌کنند و از این تعداد یک فرآیند از z عبور می‌کند و ۱۴ فرآیند روی z به خواب می‌روند.

۵- تست حذف شد.

اطلاعات مسئله ناقص است.

۶- گزینه «۴» صحیح است.

روش‌های زمان‌بندی فرآیندها هیچ تأثیری بر زمان اجرا یا همان Burst ندارد.

۷- گزینه «۲» صحیح است.

این عمل در یک روتین سرویس وقفه انجام می‌شود.