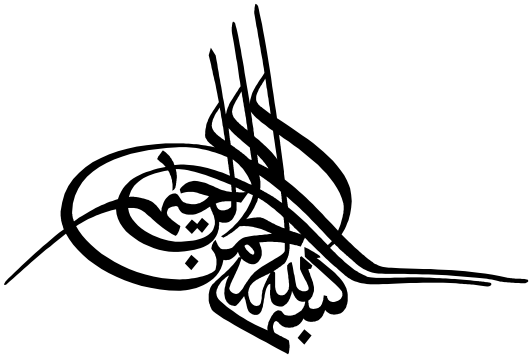


سیستم‌های عامل

سری کتاب‌های کمک آموزشی کارشناسی ارشد

مجموعه مهندسی کامپیوتر و فناوری اطلاعات

مؤلف: زهرا شولی

سرشناسه : شولی، زهرا
عنوان : سیستم‌های عامل
مشخصات نشر : تهران : مشاوران صعود ماهان، ۱۳۹۹
مشخصات ظاهری : ۱۹۷ ص
فروست : سر ی کتاب‌های کمک آموزشی کارشناسی ارشد
شابک : ۹۷۸-۶۰۰-۴۵۸-۸۱۰-۲
وضعیت فهرست نویسی : فیپای مختصر
یادداشت :این مدرک در آدرس http://opac.nlai.ir قابل دسترسی است.
شماره کتابشناسی ملی : ۳۷۶۸۲۳۶

نام کتاب: .....سیستم‌های عامل
مولف:.....زهرا شولی
ناشر:.....مشاوران صعود ماهان
مدیر تولید محتوی.....سمیه بیگی
نوبت و تاریخ چاپ:.....اول / ۱۳۹۹
تیراژ:.....۱۰۰۰ نسخه
قیمت:.....۶۶۰/۰۰۰ ریال
شابک: .....۹۷۸-۶۰۰-۴۵۸-۸۱۰-۲ :ISBN

انتشـــارات مشاوران صعود ماهان: خیابان ولیعصر، بالاتر از تقـاطع مطهری،
روبروی قنادی هتل بزرگ تهـــران، جنب بانک ملی، پلاک ۲۰۵۰
تلفن: ۴- ۸۸۱۰۰۱۱۳

سخن ناشر

«ن والقلم و ما یسطرون»

کلمه نزد خدا بود و خدا آن را با قلم بر ما نازل کرد.

به پاس تشکر از چنین موهبت الهی، موسسه ماهان درصدد برآمده است تا در راستای انتقال دانش و مفاهیم با کمک اساتید مجرب و مجموعه کتب آموزشی خود برای شما داوطلبان ادامه تحصیل در مقطع کارشناسی ارشد گام موثری بردارد. امید است تلاش‌های خدمتگزاران شما در این موسسه پایه‌گذار گام‌های بلند فردای شما باشد.

مجموعه کتاب‌های کمک آموزشی ماهان به‌منظور استفاده داوطلبان کنکور کارشناسی ارشد سراسری و آزاد تالیف شده‌اند. در این کتاب‌ها سعی کرده‌ایم با بهره‌گیری از تجربه اساتید بزرگ و کتب معتبر داوطلبان را از مطالعه کتاب‌های متعدد در هر درس بی‌نیاز کنیم.

دیگر تالیفات ماهان برای سایر دانشجویان به‌صورت ذیل می‌باشد.

- **مجموعه کتاب‌های ۸ آزمون:** شامل ۵ مرحله کنکور کارشناسی ارشد ۵ سال اخیر به همراه ۳ مرحله آزمون تالیفی ماهان همراه با پاسخ تشریحی می‌باشد که برای آشنایی با نمونه سوالات کنکور طراحی شده است. این مجموعه کتاب‌ها با توجه به تحلیل ۳ ساله اخیر کنکور و بودجه‌بندی مباحث در هریک از دروس، اطلاعات مناسبی جهت برنامه‌ریزی درسی در اختیار دانشجو قرار می‌دهد.
- **مجموعه کتاب‌های کوچک:** شامل کلیه نکات کاربردی در گرایش‌های مختلف کنکور کارشناسی ارشد می‌باشد که برای دانشجویان جهت جمع‌بندی مباحث در ۲ ماهه آخر قبل از کنکور مفید می‌باشد.

بدین‌وسیله از مجموعه اساتید، مولفان و همکاران محترم خانواده بزرگ ماهان که در تولید و به‌روزرسانی تالیفات ماهان نقش موثری داشته‌اند، صمیمانه تقدیر و تشکر می‌نماییم.

دانشجویان عزیز و اساتید محترم می‌توانند هرگونه انتقاد و پیشنهاد درخصوص تالیفات ماهان را از طریق سایت ماهان به آدرس mahan.ac.ir با ما در میان بگذارند.

موسسه آموزش عالی آزاد ماهان

سخن مولف

سالیان طولانی را در راه کسب علم و دانش پشت سر گذارده‌ایم ولی می‌دانیم که هنوز جاده دانش بس دراز است و رسیدن به سرمنزل مقصود دشوار و عمل به‌همین علم و دانش اندک دشوارتر. به‌همین‌سبب در این زمان تصمیم به نگارش و گردآوری کتابی کرده‌ایم که بتوانیم با آن چراغی باشیم به‌سمت ارتقاء هرچه بهتر علم و اندکی سختی و دشواری راه را از جان طالب علم برداریم.

ازآنجاکه یکی از دغدغه‌های دانشجویان رشته کامپیوتر و فناوری اطلاعات، ادامه تحصیل در مقاطع بالا به منظور کسب شرایط و موقعیت بهتر علمی و شغلی است این کتاب که حاصل تلاش اینجانب می‌باشد برای رفع نیاز دانشجویان و یا فارغ التحصیلانی است که در حال حاضر درصدد شرکت در آزمون کارشناسی ارشد می باشند. ما در این مجموعه سعی بر این داشته‌ایم تا مطالب و نکات مهم مبحث سیستم‌های عامل را در ۷ فصل بیان کنیم همچنین سوالات کنکور سراسری و آزاد همراه با پاسخ تشریحی را در قالب کتاب بگنجانیم.

امیدواریم مجموعه حاضر، برای داوطلبان و علاقمندان به ادامه تحصیل در مقاطع بالاتر مفید باشد، هرچند ادعایی بر بی‌نقص بودن آن نداریم. در آخر از همه دوستان و استادان گرامی تقاضا داریم اگر اشتباهی به سهو، در آن رفته، به چشم عفو بنگرند و نظرات سازنده خود را برای بهتر کردن و رفع نواقص آن با ایمیل zahrashooli@yahoo.com درمیان بگذارند.

زهرا شولی

فهرست

عنوان | صفحه

فصل اول- مقدمه و تعاریف اولیه ۹
اهداف و وظایف سیستم عامل ۱۰
سیستم‌عامل‌های دسته‌ای ۱۱
سیستم‌عامل‌های ناپیوسته ۱۲
سیستم‌عامل‌های Spool ۱۲
سیستم‌عامل‌های چندبرنامه‌ای ۱۳
سیستم‌عامل‌های محاوره‌ای ۱۳
سیستم‌عامل‌های اشتراک زمانی ۱۳
سیستم‌عامل‌های بلادرنگ ۱۳
سیستم‌عامل‌های چندپردازنده‌ای یا موازی ۱۳
سیستم‌عامل‌های توزیع‌شده و شبکه‌ای ۱۴
بخش‌های سیستم‌عامل ۱۴
فراخوانی سیستم ۱۵
روش‌های انتقال ورودی / خروجی ۱۵
دسترسی مستقیم به حافظه ۱۶
وقفه‌ها ۱۷
ساختارهای مختلف سیستم‌عامل ۲۲
سؤالات چهارگزینه‌ای و پاسخنامه سراسری فصل اول ۲۴
سؤالات چهارگزینه‌ای و پاسخنامه آزاد فصل اول ۲۶
فصل دوم- فرآیندها و نخ‌ها ۲۹
ایجاد و پایان فرآیند ۳۰
وضعیت‌های یک فرآیند ۳۱
حالات اجرا ۳۳
مدیریت فرآیند حافظه / ورودی ـ خروجی ۳۴
بلوک کنترلی فرآیند ۳۴
رابطه بین نخ‌ها و فرآیندها ۳۷
حالات نخ ۳۷
سؤالات چهارگزینه‌ای و پاسخنامه سراسری فصل دوم ۳۸
سؤالات چهارگزینه‌ای و پاسخنامه آزاد فصل دوم ۴۴

فصل سوم- زمان‌بندی فرآیندها ۴۵
زمان‌بندی برحسب ورودی (FIFO or FCFS) ۴۷
زمان‌بندی گردشی نوبتی (RR) ۴۹
زمان تعویض متن ۵۰
روش زمان‌بندی برحسب کوچک‌ترین زمان سرویس (SPN or SJF) ۵۱
تعیین زمان سرویس ۵۲
روش زمان‌بندی برحسب کوچک‌ترین زمان باقیمانده (SRT) ۵۲
روش زمان‌بندی برحسب کوچک‌ترین زمان پاسخ (HRRN) ۵۳
روش زمان‌بندی قرعه‌کشی (Lottery) ۵۳
روش نوبتی گردش مجازی (VRR) ۵۴
روش زمان‌بندی بازخوردی (FBS) ۵۴
روش زمان‌بندی سهمیه‌بندی عادلانه (FSS) ۵۴
روش زمان‌بندی برحسب موعد مقرر (DS) ۵۴
سؤالات چهارگزینه‌ای و پاسخنامه سراسری فصل سوم ۵۵
سؤالات چهارگزینه‌ای و پاسخنامه آزاد فصل سوم ۶۶
فصل چهارم- ناحیه بحرانی و مقابله با شرایط مسابقه ۷۱
پردازش موازی یا همروند (Concurrent Process) ۷۲
شرایط مسابقه (Race condition) ۷۴
الگوریتم دکر (Dekker’s Algorithm) ۷۵
انتظار مشغولی (Busy waiting) ۷۸
الگوریتم پترسن (Peterson’s Algorithm) ۸۰
راه‌حل‌های سخت‌افزاری برای حل مساله ناحیه بحرانی ۸۱
پدیده وارونگی اولویت (Priority Inversial) ۸۳
راهنما یا سمافور (Semaphore) ۸۴
چند مثال از استفاده از سمافور باینری برای حل مسائل ناحیه بحرانی ۸۵
روش‌های کامپایلری برای حل مساله ناحیه بحرانی ۸۸
بررسی مساله خوانندگان و نویسندگان ۹۲
سؤالات چهارگزینه‌ای و پاسخنامه سراسری فصل چهارم ۹۶
سؤالات چهارگزینه‌ای و پاسخنامه آزاد فصل چهارم ۱۱۲
فصل پنجم- بن‌بست و راه‌های مقابله با آن ۱۲۱
شرایط رویداد بن‌بست ۱۲۳
گراف وضعیت سیستم ۱۲۳
راه‌های مقابله با بن‌بست ۱۲۴
الگوریتم تشخیص بن‌بست ۱۲۴
الگوریتم بانکداری ۱۲۵
مرحله ترمیم بن‌بست ۱۲۶
اجتناب از بن‌بست ۱۲۷

پیشگیری از بن‌بست ۱۲۸
قحطی‌زدگی (Starvation) ۱۲۹
سؤالات چهارگزینه‌ای و پاسخنامه سراسری فصل پنجم ۱۳۰
سؤالات چهارگزینه‌ای و پاسخنامه آزاد فصل پنجم ۱۳۵
فصل ششم- مدیریت حافظه ۱۳۷
روش مدیریت تک‌برنامه‌ای ۱۳۸
روش مدیریت حافظه پارتیشن‌بندی ثابت ۱۳۹
مساله جابجایی ۱۳۹
روش مدیریت حافظه پارتیشن‌بندی متغیر ۱۴۰
روش حسابداری حافظه Bitmap ۱۴۱
روش حسابداری حافظه لیست پیوندی ۱۴۱
روش مدیریت حافظه صفحه‌بندی ۱۴۳
روش قطعه‌بندی ۱۴۵
جدول صفحه چندسطحی ۱۴۷
جدول صفحه معکوس ۱۴۹
جدول درهم‌سازی ۱۴۹
تابع درهم‌سازی ۱۵۰
الگوریتم جستجو درجدول درهم‌سازی ۱۵۱
میانگیر دم‌دستی ترجمه ۱۵۱
ترکیب صفحه‌بندی و قطعه‌بندی ۱۵۳
روش روی‌هم‌گذاری یا هم‌پوشانی یا جای‌گذاشت ۱۵۴
الگوریتم‌های جایگزینی صفحات ۱۵۵
الگوریتم LFU ۱۵۸
الگوریتم MFU ۱۵۸
الگوریتم NRU ۱۵۹
تناقض بلیدی ۱۶۰
سؤالات چهارگزینه‌ای و پاسخنامه سراسری فصل ششم ۱۶۱
سؤالات چهارگزینه‌ای و پاسخنامه آزاد فصل ششم ۱۷۴
فصل هفتم- مدیریت دیسک ۱۸۱
انواع مختلف دیسک‌ها ۱۸۳
الگوریتم‌های زمان‌بندی دیسک ۱۸۴
درهم‌چینی یا تورق ۱۸۶
سؤالات چهارگزینه‌ای و پاسخنامه سراسری فصل هفتم ۱۸۸
سؤالات چهارگزینه‌ای و پاسخنامه آزاد فصل هفتم ۱۹۲
سوالات و پاسخ کنکور سراسری ۹۵ ۱۹۳
منابع ۱۹۷

فصل اول

مقدمه و تعاریف اولیه

- اهداف سیستم‌عامل
- وظایف سیستم‌عامل
- روند تکامل سیستم‌عامل‌ها
- وقفه‌ها
- ساختارهای مختلف سیستم‌عامل
- جداول سیستم‌عامل

مقدمه و تعاریف اولیه

سیستم‌عامل: برنامه‌ای است که برای اجرای برنامه‌های کاربردی و کنترل آنها طراحی می‌گردد و رابط بین کاربر و سخت‌افزار می‌باشد.

اهداف سیستم‌عامل

۱) استفاده از کامپیوتر توسط سیستم‌عامل ساده‌تر و راحت‌تر شود. (سهولت)
۲) با استفاده از سیستم‌عامل می‌توان از منابع کامپیوتر به‌صورت کارآمدتری استفاده نمود. (کارآمدی)
۳) طراحی سیستم‌عامل باید به گونه‌ای باشد که بتوان قابلیت جدید را بدون اخلال در خدمات جاری در آن تعریف نمود. (قابلیت رشد)

پس به‌طور خلاصه می‌توان گفت که سه هدف از طراحی سیستم‌عامل عبارتند از:

۱) مخفی نگه داشتن پیچیدگی‌های سخت‌افزار از کاربر
۲) استفاده بهینه از منابع سیستم
۳) کنترل اجرای برنامه‌ها و جلوگیری از خطا و امکان رشد و توسعه

لایه‌های یک سیستم کامپیوتری در شکل زیر نشان داده شده است. در این سیستم لایه‌بندی کاربر در بالاترین لایه و سخت‌افزار کامپیوتر در پایین‌ترین لایه می‌باشد. این کار برای به وجود آوردن سهولت بیشتر و امنیت بالاتر می‌باشد. در این لایه‌بندی کاربر، کارهای خود را توسط برنامه‌های کاربردی انجام می‌دهد. برنامه‌های کاربردی برای کار با حافظه، سخت‌افزار و غیره از روتین‌های سیستم بهره می‌گیرند. این برنامه‌ها دستورات کاربر را به سیستم‌عامل می‌برند و از آنجا سیستم‌عامل آنها را بررسی می‌کند و بعد از تبدیل دستورات به ریزبرنامه آنها را به سخت‌افزار کامپیوتر می‌فرستد.

وظایف سیستم‌عامل

به‌طور خلاصه وظایف سیستم‌عامل را می‌توان به‌صورت زیر بیان کرد:

۱) توسعه برنامه‌ها: با ارائه امکاناتی نظیر ویرایشگر متنی و اشکال‌زداها، سیستم‌عامل سعی می‌کند که به روند شکل‌گیری برنامه‌ها کمک کند.

۲) اجرای برنامه‌ها: وظیفه بار کردن برنامه‌ها بر روی حافظه، انجام خدمات I/O و کار کردن با فایل‌ها بر عهده سیستم‌عامل می‌باشد.
۳) کنترل دسترسی به پرونده‌ها
۴) به اشتراک گذاشتن سخت‌افزار بین کاربرها
۵) به اشتراک گذاشتن داده‌ها بین کاربرها
۶) برآورده کردن امکانات خواندن و نوشتن
۷) سازماندهی داده‌ها، دستیابی سریع و مطمئن
۸) ترمیم خطا
۹) کنترل دستیابی به منابع
۱۰) حسابداری استفاده از منابع: یعنی در هر لحظه می‌داند که کدام فرآیند با کدام منبع کار می‌کند و کدام منبع آزاد است.
۱۱) امکان پردازش موازی: منظور از پردازش موازی در اینجا کار کردن چند کاربر به‌طور هم‌زمان می‌باشد.

روند تکامل سیستم‌عامل‌ها

نکته: روند تکامل سیستم‌عامل همگام با روند تکامل سخت‌افزار انجام شده است. یعنی هرگاه امکانات سخت‌افزاری رشد کرده‌اند، سیستم‌عامل‌ها نیز پیشرفت کرده‌اند.

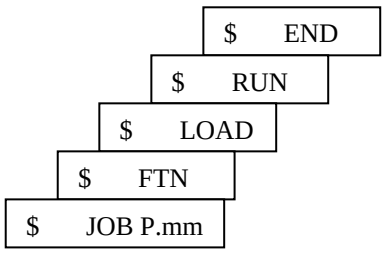
از آنجایی‌که سیستم‌عامل‌ها از نظر تاریخی بسیار به کامپیوتری که روی آن اجرا می‌شوند مربوط هستند، به تکامل نسل کامپیوترها می‌پردازیم تا ببینیم در هر نسل سیستم‌عامل‌ها چگونه کار می‌کردند.

اولین کامپیوتر دیجیتال توسط ریاضی‌دان انگلیسی Charles – Babbage (۱۷۹۲-۱۸۷۱) طراحی شد. گرچه ببیج بیشتر وقت و سرمایه خود را روی ساخت موتور تحلیلی (Analizing engine) گذاشت اما هیچ‌گاه نتوانست به‌طور کامل آن را طراحی کند زیرا صرفاً مکانیکی بود و تکنولوژی آن روزگار نمی‌توانست چرخ‌ها و چرخ دنده‌هایی که او نیاز داشت را با دقت کافی در اختیارش قرار دهد. واضح است که موتور تحلیلی هیچ سیستم‌عاملی نداشت.

ماشین‌های نسل اول، سیستم‌عامل نداشتند. اولین سیستم‌عامل در اوایل سال ۱۹۵۰ توسط شرکت جنرال موتورز طراحی شد و اسم آن Monitor بود که از کارت پانچ‌ها استفاده می‌کرد و خواسته‌ها را روی کارت (Job) پانچ می‌کرد. هرگاه Job ای آماده می‌شد آن را به اپراتور می‌دادند و او توسط دستگاه کارت ریدر آن‌ها را می‌خواند و نتیجه را چاپ می‌کرد.

نکته: پس اولین سیستم‌عامل‌ها تک کاربره بودند.

در این سیستم‌ها به هر بخش از Job، قدم (Step) می‌گفتند. نمونه‌ای از یک Job در زیر آمده است:



سیستم‌عامل‌های دسته‌ای (BATCH)

این نوع سیستم‌عامل‌ها می‌توانستند کارها را به‌صورت دسته‌ای اجرا کنند و مطمئناً خیلی سریع‌تر از سیستم‌عامل‌های قبلی بودند.

ایده اصلی در ورای طرح پردازش دسته‌ای ساده، استفاده از نرم‌افزاری به نام «ناظر» است. با استفاده از سیستم‌عامل دسته‌ای، دیگر کاربر دسترسی مستقیم به ماشین ندارد. در عوض، کاربر کار خود را روی کارت یا نوار به متصدی می‌دهد. اپراتور کارها را به‌صورت ردیفی دسته کرده و همگی را روی دستگاه ورودی می‌گذارد تا مورد استفاده ناظر قرار گیرد. هر برنامه به نوعی ساخته

شده که با تکمیل پردازش به ناظر انشعاب کند و ناظر به‌طور خودکار، بار کردن برنامه بعدی را آغاز کند. در واقع مساله زمان‌بندی به عهده ناظر است. مجموعه یا بسته‌ای از کارها به صف شده و بدون اتلاف وقت با بیشترین سرعت اجرا می‌شوند. زمان تنظیم شرایط اولیه کارها را نیز ناظر بهتر می‌کند. با هرکار دستورالعمل‌هایی از زبان کنترل کار (JCL) نیز وجود دارد. زبان کنترل کار، نوعی زبان برنامه نویسی برای فرمان دادن به ناظر می‌باشد.

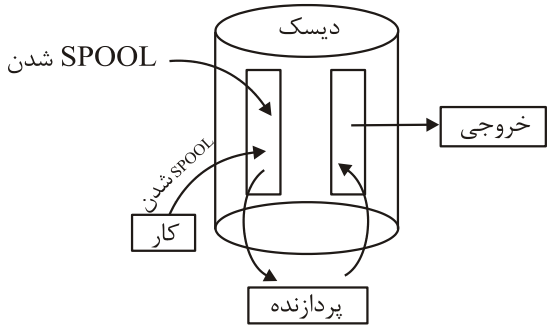
سیستم‌عامل‌های ناپیوسته (Offline)

سیستم‌عامل‌هایی بودند که در آنها بین تولید داده و پردازش آنها تأخیر وجود داشت. مهم‌ترین اشکال این سیستم‌عامل‌ها این بود که در یک زمان یا عمل خواندن و یا عمل نوشتن قابل انجام بود و انجام هم‌زمان هر دو عمل امکان‌پذیر نبود.

سیستم‌عامل‌های SPOOL (SPOOL: Simultanous Peripheral Operating Online)

در این سیستم‌عامل‌ها برای استفاده بهینه از وقت پردازنده، در زمانی که یک کار اجرا می‌شود کار دیگر بر روی دیسک بار می‌شود (یا اصطلاحاً می‌گوییم SPOOL می‌شود).

نکته: اگر کار روی نوار (tape) ذخیره شود اصطلاحاً به جای SPOOL از کلمه Stage استفاده می‌کنیم.



نکته: سیستم‌عامل Spool با یک کامپیوتر انجام می‌شود ولی سیستم‌عامل Offline با چند کامپیوتر انجام می‌شود (در این‌گونه سیستم‌عامل‌ها، از یک کامپیوتر ضعیف برای تبدیل Jobها به tape و یک کامپیوتر قوی برای اجرای برنامه‌های روی tape استفاده می‌شود.

نکته: سیستم‌عامل‌های دسته‌ای کارها را به‌صورت دسته‌ای دریافت می‌کند و به ترتیب اجرا می‌کنند، بنابراین زمان تلف شده بین انجام کارها از بین می‌رود و سرعت سیستم بالاتر می‌رود و کارآیی پردازنده نیز بیشتر می‌شود. همچنین این سیستم‌عامل‌ها با ایجاد دسته کارهای هم نوع در زمان، صرفه جویی می‌کنند. مثلاً دسته برنامه‌هایی را در نظر بگیرید که همگی نیاز به کامپایلر دارند. این سیستم‌عامل‌ها همه آنها را دسته‌بندی می‌کند و با یک بار اجرا کردن کامپایلر همه آنها را انجام می‌دهد.

نکته: دلیل استفاده از سیستم‌عامل‌های Offline سرعت بسیار کم کارت خوان‌ها بود. بنابراین در آنها از یک کامپیوتر ارزان قیمت برای تبدیل کارت‌ها به نوار استفاده می‌شد و سپس نوار را به کامپیوتر اصلی می‌دادند. این سیستم‌عامل‌ها را می‌توان دسته‌ای هم در نظر گرفت، زیرا می‌توانستیم چندین کار مختلف را روی یک نوار قرار دهیم.

نکته: تفاوت Spool و Buffer

۱- Spool با درخواست کاربر پر می‌شود ولی Buffer قبل از آنکه کاربر درخواستی را اعلام کند، پر می‌شود و در مواقع نیاز به کاربر تحویل می‌گردد. مثلاً بافر خروجی سبب می‌شود که سرعت خواندن و نوشتن بالا رود و وقفه‌ای روی ندهد. یا مثلاً وقتی به موسیقی گوش می‌دهید تا چند ثانیه بعد از لحظه جاری بافر می‌شود تا صدا با وقفه پخش نشود.

۲ـ Spool در واقع حافظه جانبی است و هم‌زمانی بین دو کار مختلف را سبب می‌شود. یعنی در حالی که یک کار اجرا می‌شود، کار دیگر روی دیسک Spool می‌شود ولی Buffer در واقع قسمتی از حافظه اصلی است و هم‌زمانی یک کار با خودش را سبب می‌شود. (یعنی در هنگام اجرای یک کار اطلاعاتی که در آینده نزدیک به آنها نیاز دارد، بافر می‌شود.)

سیستم‌عامل‌های چند برنامه‌ای (Multi Programming OS)

معمولاً در سیستم‌های تک برنامه‌ای، پردازنده زمان زیادی را بی‌کار است. چون زمان زیادی صرف کار کردن با I/O می‌شود. برای جلوگیری از این امر، سیستم‌عامل‌های چند برنامه‌ای طراحی شدند. در این گونه سیستم‌عامل‌ها، در یک زمان چند برنامه در حال اجرا هستند و پردازنده بین آنها دست به دست می‌شود.

سیستم‌عامل‌های محاوره‌ای (Intractive Operating Systems)

این نوع سیستم‌عامل‌ها تک کاربره هستند و بر خلاف سیستم‌عامل‌های Batch که کاربر تا پایان اجرای برنامه از وضعیت آن مطلع نبود و نمی‌توانست متوجه شود که آیا خطایی رخ داده است یا خیر، سیستم‌عامل‌های محاوره‌ای در هر لحظه با کاربر در ارتباط هستند.

سیستم‌عامل‌های اشتراک زمانی (Time Sharing Operating Systems)

این نوع سیستم‌عامل‌ها نیز محاوره‌ای هستند با این تفاوت که چند کاربره هستند و هرکاربر توسط یک ترمینال با سیستم در ارتباط است و کارهای خود را اجرا می‌کند.

سیستم‌عامل‌های بلادرنگ (Real Time Operating Systems)

از آنجا که امروزه کاربرد کامپیوترها در همه امور روزمره رایج شده است و از کامپیوترها در مراکز حساس از جمله سیستم کنترل پرواز و... استفاده می‌گردد، نیاز به سیستم‌عامل‌هایی که قابل اعتماد باشند و در زمان کم پاسخگو باشند زیاد شده است. به این سیستم‌عامل‌ها بلادرنگ گویند. در واقع در این سیستم‌ها زمان نقش حیاتی دارد و اگر سیستم در زمانی مشخص و تعیین شده پاسخ ندهد با عدم اجرا یکسان است مثل سیستم‌های کنترل موشک، کنترل صنعتی و غیره.

سیستم‌عامل‌های چند پردازنده‌ای یا موازی

در این نوع سیستم‌ها چندین پردازنده وجود دارندکه می‌توانند به‌صورت موازی چندین فرآیند یا چندین نخ از یک فرآیند را اجرا کنند مزایای این سیستم‌ها عبارتند از:

- از آنجا که در این سیستم‌ها همه پردازنده‌ها به‌صورت اشتراکی از حافظه‌ها، وسایل جانبی، منابع تغذیه و غیره استفاده می‌کنند بنابراین در هزینه‌های سخت‌افزاری صرفه‌جویی می‌شود.
- چون با وجود چندین پردازنده سرعت بسیار بالا می‌رود بنابراین توان عملیاتی یا throughput افزایش می‌یابد. یعنی تعداد کارهایی که در یک پریود زمانی کامل می‌شوند افزایش می‌یابد.
- قابلیت اعتماد در این سیستم‌ها به‌دلیل تحمل پذیری آنها در برابر خطا افزایش می‌یابد یعنی خرابی یک پردازنده باعث متوقف شدن سیستم نخواهد شد فقط سرعت سیستم را کمتر خواهد کرد.

این نوع سیستم‌ها به دو دسته زیر تقسیم می‌شوند:

۱) چند پردازنده‌ای متقارن (SMP: Symmetric Multi Programming)

۲) چند پردازنده‌ای نامتقارن (ASMP: Asymmetric Multi Programming)

در سیستم‌های چند پردازنده‌ای SMP هر پردازنده می‌تواند هر کاری را انجام دهد. ولی در سیستم‌های چند پردازنده‌ای ASMP هر پردازنده مسئول انجام وظیفه از قبل تعیین شده‌ای است معمولاً به یکی از پردازنده‌ها Master و به بقیه Slave گویند.

سیستم‌های SMP کارآیی بالاتری دارند. اما انتخاب طراحی ASMP یا SMP به طراحی سیستم و سخت‌افزار بستگی دارد.

سیستم‌عامل‌های Windows، Linux و Solaris به‌طور معمول SMP هستند اما ورژن‌هایی از همین سیستم‌عامل‌ها به‌صورت ASMP کار می‌کنند.

نکته: طراحی سیستم‌های متقارن از پیچیدگی بیشتری برخوردار است. سیستم‌های چند پردازنده‌ای به نام Cluster وجود دارند که می‌توانند متقارن یا نامتقارن باشند. این سیستم‌ها از حافظه مشترک استفاده می‌کنند و سیستم‌های جدیدی هستند که در مرحله تحقیق به سر می‌برند.

سیستم‌عامل‌های توزیع‌شده و شبکه‌ای

در سال ۱۹۸۰ سیستم‌عامل‌های شبکه‌ای ابداع شدند که Login از راه دور بودند. سیستم‌عامل‌های توزیع شده (Distributed Operating Systems) در سال ۱۹۹۰ پا به عرصه گذاشتند در آنها نیز سخت‌افزار شکل شبکه‌ای داشت. در این سیستم‌عامل‌ها وقتی کاربر پشت کامپیوتر می‌نشیند درست مانند این است که پشت یک Terminal نشسته است و وقتی برنامه‌ای را save می‌کند، نمی‌داند در کجا save می‌شود و نمی‌داند از کدام پردازنده استفاده می‌کند.

تفاوت‌های سیستم عامل‌های توزیع شده و شبکه:

- در شبکه کاربران از وجود ماشین‌های مختلف در شبکه خبر دارند اما در سیستم توزیع شده این موضوع از دید کاربران مخفی می‌ماند.
- در شبکه هر کامپیوتر می‌تواند سیستم‌عامل ویژه خود را داشته باشد و تنها یک نرم‌افزار مدیریت شبکه به سیستم‌عامل اضافه شده است ولی در سیستم توزیع شده تنها یک سیستم‌عامل مجموعه‌ای از سیستم‌های متصل به هم را مدیریت می‌کند.

سیستم‌عامل‌های توکار (Embeded Operating Systems) سیستم‌عامل‌های کوچکی هستند که در ابزارهایی مانند موبایل، آسانسور و... از آنها استفاده می‌شود. از ویژگی‌های این سیستم‌عامل‌ها، حجم حافظه بسیار کوچک آنها می‌باشد. Windows CE یا سیستم‌عامل‌هایی که روی تراشه‌های کارت‌های اعتباری قرار دارند، نمونه‌ای از این سیستم‌عامل‌ها می‌باشند.

بخش‌های سیستم‌عامل

سیستم‌عامل از تعدادی مدیریت یا بخش تشکیل شده است که هر بخش مسئول انجام وظیفه خاصی می‌باشد و این بخش‌ها عبارتند از:

۱) مدیریت پردازه‌ها یا فرآیندها

وظایف این بخش عبارتند از: تولید و حذف فرآیندها- اختصاص پردازنده به فرآیندها- نگهداری اطلاعات درباره فرآیند (اطلاعاتی چون وضعیت فرآیندها، ارتباط بین فرآیندها، همگام‌سازی فرآیندها) جلوگیری از وقوع بن‌بست.

۲) مدیریت حافظه اصلی

یکی از مهم‌ترین منابع کامپیوتری، حافظه اصلی می‌باشد. هر فرآیند برای اجرا شدن، ابتدا باید در حافظه اصلی مستقر شود. مدیریت حافظه باید حافظه را اداره کند و به‌صورت مستمر فرآیندهایی را به حافظه بیاورد و فرآیندهایی که کارشان تمام شده است را از حافظه بیرون ببرد تا فرآیند دیگری بتواند در حافظه قرار گیرد. نگهداری اطلاعات درباره این که کجای حافظه خالی و کجای آن اشغال است و اینکه چه هنگام به فرآیند حافظه اختصاص داده شود از وظایف این بخش می‌باشد.

۳) مدیریت حافظه جانبی

تمامی برنامه‌ها در ابتدا روی حافظه جانبی مستقر هستند زیرا حافظه اصلی آنقدر گنجایش ندارد که بتواند همه برنامه‌ها را در خود جای دهد. زمان‌بندی دیسک (حافظه جانبی) و اختصاص حافظه از وظایف این بخش می‌باشد.

۴) مدیریت فایل‌ها یا پرونده‌ها

فایل‌ها یکی دیگر از منابع کامپیوتر می‌باشد و از وظایف این بخش، سازماندهی فایل‌ها، تولید و حذف دایرکتوری تولید و حذف فایل‌ها و تخصیص فایل‌ها به فرآیند و پس گرفتن آنها می‌باشد. همچنین مسئولیت حفاظت از اطلاعات فایل‌ها نیز بر عهده این بخش از سیستم‌عامل می‌باشد.

۵) مدیریت دستگاه‌های ورودی و خروجی

این بخش از سیستم‌عامل مسئول اجرای دستورات I/O ای است که فرآیندها در طول اجرای خود به آنها نیاز دارند و باعث می‌شود که کاربر متوجه پیچیدگی دستگاه‌های ورودی و خروجی نشود. از جمله وظایف این بخش بافر کردن فایل‌ها می‌باشد.

۶) حفاظت و مفسر دستورات

تمام مدیریت‌هایی که در فوق آمدند یک بخش حفاظت نیز در خود دارند که از جمله آنها می‌توان حفاظت از فرآیندها، حافظه اصلی، حافظه جانبی، فایل‌ها و I/O را نام برد.

مفسر دستورات (Command Interpreter): این برنامه، یک برنامه واسط بین کاربر و سیستم‌عامل می‌باشد و دستورات کاربر را خط به خط برای سیستم‌عامل تفسیر می‌کند. البته بعضی از CIها گرافیکی هستند (مانند Windows) که در آن‌ها دستورات می‌توانند توسط حرکت ماوس تولید شوند. مثلاً Lindows یک CI جدید است از linux که محیط آن را بسیار شبیه به windows کرده است.

فراخوانی سیستم (System Call)

سیستم‌عامل از تعدادی روتین (routine) تشکیل شده و معمولاً به کاربر اجازه داده می‌شود که از درون برنامه خود آنها را صدا کند و از آنها استفاده کند. این کار از طریق فراخوانی سیستم (System Call) صورت می‌گیرد. فراخوانی سیستم سبب ایجاد یک وقفه (inttrupt) می‌شود. (مثلاً در اسمبلی دستورات int 10 یا int 21 دو فراخوانی سیستم می‌باشند. همچنین در زبان C، Cin یک فراخوانی سیستم است.).

روش‌های انتقال ورودی/ خروجی

برای عملیات ورودی/ خروجی سه روش وجود دارد:

ورودی/ خروجی برنامه‌سازی شده. (یا روش سرکشی polling)

ورودی/ خروجی مبتنی بر وقفه

دسترسی مستقیم به حافظه (DMA)

ورودی/ خروجی برنامه‌سازی شده (سرکشی یا polling):

هنگامی‌که پردازنده در حال اجرای یک برنامه است با مواجهه با یک دستورالعمل مرتبط با ورودی/ خروجی، فرمان لازم را به مولفه ورودی/ خروجی مربوط صادر می‌کند. با ورودی/ خروجی برنامه‌سازی شده، مولفه ورودی/ خروجی، عمل درخواست شده را انجام داده و بیت‌های مناسب از ثبات وضعیت ورودی/ خروجی را مقدارگذاری می‌کند. (ثبات وضعیت یا psw، ثباتی است که در آن اطلاعات مربوط به یک پردازش از دیدگاه سخت‌افزاری مثل پرچم‌ها، بیت‌های کنترل وقفه و بردار سرویس‌دهی به وقفه ذخیره می‌شود به عبارت دیگر ثبات وضعیت یک ثبات کنترلی است که پردازنده با بررسی آن می‌تواند متوجه شود که آیا عملیات انتقال

یک دستگاه ورودی/ خروجی تمام شده یا نه). مجموعه دستورالعمل‌های پردازنده در روش ورودی/ خروجی برنامه‌سازی شده شامل گروه‌های دستورالعمل‌های ورودی/ خروجی زیر می‌باشد:

کنترل: دستورالعمل‌هایی برای فعال کردن دستگاه خارجی و اینکه چه باید انجام دهد. مثلا ممکن است به دستگاه نوار مغناطیسی فرمان داده شود که به ابتدای نوار برگردد و یا یک رکورد جلو رود.

وضعیت: دستورالعمل‌هایی برای بررسی وضعیت مولفه‌های ورودی/ خروجی و دستگاه‌های جنبی آن.

انتقال: دستورالعمل‌هایی برای انتقال داده‌ها بین ثبات‌های پردازنده و دستگاه‌های خارجی به روش ورودی/ خروجی برنامه‌سازی شده، روش انتظار مشغول (Busy waiting) هم گفته می‌شود چرا که وقت زیادی از پردازنده به‌خاطر سرکشی‌های متناوب هدر می‌رود.

ورودی/ خروجی مبتنی بر وقفه

مشکل روش ورودی/ خروجی برنامه‌سازی شده این است که پردازنده باید برای مدت طولانی منتظر بماند تا مولفه ورودی/ خروجی برای دریافت یا ارسال آماده شود. در مدت انتظار، پردازنده باید مکررا وضعیت مولفه ورودی/ خروجی را مورد سوال قرار دهد، در نتیجه کارایی سیستم به شدت پایین می‌آید.

راه دیگر این است که بعد از صدور فرمان به مولفه ورودی/ خروجی، پردازنده به‌کار مفید دیگری بپردازد. حال وقتی که مولفه ورودی/ خروجی برای مبادله داده‌ها آماده شد، برای درخواست خدمت به پردازنده وقفه می‌دهد. پردازنده مثل گذشته انتقال داده‌ها را انجام داده و بعد پردازش قبلی خود را از سر می‌گیرد. به این روش ورودی/ خروجی مبتنی بر وقفه می‌گویند.

دسترسی مستقیم به حافظه (DMA) (Direct Memory access)

گرچه ورودی/ خروجی مبتنی بر وقفه از ورودی/ خروجی برنامه‌سازی شده کارآمدتر است، اما هنوز نیاز به دخالت فعال پردازنده برای انتقال داده‌ها بین حافظه و مولفه ورودی/ خروجی است و هرگونه انتقالی، مسیری از داخل پردازنده را طی می‌کند، دو روش ورودی/ خروجی دو اشکال ذاتی دارند:

۱- نرخ انتقال ورودی/ خروجی محدود به‌سرعتی است که پردازنده می‌تواند یک دستگاه را بررسی کرده و خدمت دهد.

۲- پردازنده گرفتار مدیریت انتقال ورودی/ خروجی است.

برای انتقال داده‌های حجیم روش کارآمدتری لازم است، دسترسی مستقیم به حافظه (DMA) عمل دسترسی مستقیم به حافظه می‌تواند به وسیله مولفه دیگری روی گذرگاه سیستم انجام شود، یا اینکه به عهده یک مولفه ورودی/ خروجی گذاشته شود. در هردو حالت، این روش به این صورت کار می‌کند که وقتی پردازنده می‌خواهد بلوکی از داده‌ها را بخواند یا بنویسد، با ارسال اطلاعات زیر به مولفه DMA به آن یک فرمان می‌دهد:

- اینکه عمل خواندن مورد نیاز است یا نوشتن.
- آدرس دستگاه ورودی/ خروجی درگیر.
- محلی از حافظه که از آن محل خواندن یا نوشتن باید شروع شود.
- تعداد کلماتی که باید خوانده یا نوشته شود.

سپس پردازنده به‌کارهای دیگر می‌پردازد. این عمل، ورودی/ خروجی را به مولفه DMA واگذار کرده است و آن مولفه مراقبت لازم را خواهد کرد. مولفه DMA یک بلوک کامل از داده‌ها (هربار یک کلمه) را مستقیما به / از حافظه منتقل می‌کند، بدون اینکه داده‌ها از پردازنده عبور کنند. وقتی که انتقال کامل شد مولفه DMA علامت وقفه‌ای برای پردازنده می‌فرستد. درنتیجه پردازنده فقط در شروع و پایان انتقال داده‌ها درگیر است و در بقیه مواقع دخالتی ندارد.

توجه شود که حتی برای ورودی/ خروجی‌های چند کلمه‌ای، DMA بسیار کارآمدتر از روش‌های ورودی/ خروجی برنامه‌سازی شده و مبتنی بر وقفه است.

وقفه‌ها

اگر در یک سیستم کامپیوتری وقفه وجود داشته باشد، هنگام عمل I/O، پردازنده می‌تواند به کار خود ادامه دهد و بعد از اتمام عمل I/O وسیله ورودی/ خروجی یک وقفه می‌فرستد و توسط آن پردازنده را از اتمام عمل مطلع می‌کند.

وقفه

هر رویدادی که سبب شود CPU اجرای عادی یک برنامه را قطع کند وقفه نامیده می‌شود، وقفه‌ها جزء مهمی از معماری کامپیوتر هستند و نحوه عملکرد آن‌ها از ماشینی به ماشین دیگر ممکن است متفاوت باشد، وقفه، راهکاری را فراهم می‌سازد تا اجرای دستورالعمل‌های جاری پردازنده موقتاً متوقف شده و دستورات سرویس‌دهی دیگری اجرا گردد و پس از آن کنترل دوباره به برنامه وقفه داده شده باز گردد. یک برنامه‌نویس اسمبلی با صدور وقفه‌های نرم‌افزاری می‌تواند به‌طور موثری با دستگاه‌های جانبی ارتباط برقرار کند.

گاهی اوقات جریان عادی اجرای یک برنامه برای پردازش رویدادی که نیاز به پاسخ سریع دارد متوقف می‌شود، سخت‌افزار کامپیوتر برای مدیریت این رویدادها مکانیسمی به نام وقفه (interrupt) را دارد.

مثال: وقتی mouse حرکت می‌کند، سخت‌افزار mouse برنامه جاری را متوقف می‌کند تا حرکت mouse گرفته شود (برای حرکت مکان‌نمای mouse روی صفحه نمایش).

روتین وقفه

وقتی CPU یک سیگنال وقفه را تشخیص می‌دهد، فعالیت جاری خود را متوقف می‌کند و روتین خاصی را فراخوانی می‌کند که روتین وقفه (interrupt handler) نام دارد، این روتین علت وقوع وقفه را تشخیص می‌دهد و عکس‌العمل مناسب را انجام می‌دهد.

بیشتر روتین‌های وقفه بعد از پایان یافتن کنترل اجرا را به برنامه متوقف شده بازمی‌گردانند، آن‌ها کلیه مقادیر ثبات‌ها را به وضعیت قبل از تولید وقفه برمی‌گردانند. بنابراین برنامه متوقف شده به گونه‌ای به اجرا ادامه می‌دهد که گویا هیچ اتفاقی نیفتاده است به جز این که سیکل‌های CPU را از دست می‌دهند. وقتی دو یا چند وقفه هم‌زمان با هم اتفاق می‌افتند، CPU از سیستم اولویت استفاده می‌کند و می‌تواند در طی اجرای بخش بحرانی یک برنامه وقفه‌ها را غیر فعال کند. وقتی دارد یک روتین وقفه را اجرا می‌کند، کلیه وقفه‌های با اولویت کمتر یا، تا زمان خاتمه اجرای روتین، غیر فعال هستند.

سیستم‌عامل با کنترل و زمان‌بندی مناسب پردازه‌ها زمینه استفاده از پردازنده را برای آنان، فراهم می‌نماید. در سیستم‌های " تک- کاره"، سیستم زمان‌بندی بسیار روشن و مشخص است، در چنین مواردی، سیستم‌عامل امکان اجرای برنامه را فراهم و صرفاً در زمانی که کاربر اطلاعاتی را وارد و یا سیستم با وقفه‌ای برخورد نماید، روند اجرا متوقف خواهد شد. وقفه، سیگنال‌های خاص ارسالی توسط نرم‌افزار و یا سخت‌افزار برای پردازنده می‌باشند، در چنین مواردی منابع صادرکننده وقفه درخواست برقراری یک ارتباط زنده با پردازنده برای اخذ سرویس و یا سایر مسائل به وجود آمده، را می‌نمایند، در برخی حالات سیستم‌عامل پردازه‌ها را با یک اولویت خاص زمان‌بندی می‌نماید. در چنین حالتی هر یک از پردازه‌ها با توجه به اولویت نسبت داده شده به آنان، قادر به استفاده از زمان پردازنده خواهند بود. در این‌چنین موارد، در صورت بروز وقفه، پردازنده آن‌ها را نادیده گرفته و تا زمان عدم تکمیل عملیات موردنظر توسط پردازنده، فرصت پرداختن به وقفه‌ها وجود نخواهد داشت. بدیهی است با توجه به نحوه برخورد پردازنده (عدم توجه به وقفه‌ها)، در سریع‌ترین زمان ممکن عملیات و فعالیت جاری پردازنده به اتمام خواهد رسید. برخی از وقفه‌ها با توجه به اهمیت خود (نظیر بروز اشکال در حافظه و یا سایر موارد مشابه)، قابل اغماض توسط پردازنده نبوده و می‌بایست صرف‌نظر از نوع و اهمیت فعالیت جاری، سریعاً به وقفه ارسالی پاسخ مناسب را ارائه گردد.

انواع وقفه

- **وقفه‌های برنامه program check**: که به دلیل اجرای بعضی دستورات رخ می‌دهند، مثلاً سرریز شدن محاسباتی، تقسیم بر صفر، اجرای دستورالعمل غیرمجاز، رجوع به آدرس خارج از محدوده مجاز کاربر. به این وقفه‌ها اغلب Trap یا تله گفته می‌شود.
- **وقفه‌های زمان‌سنج (Timer)**: این وقفه به سیستم‌عامل امکان می‌دهد بعضی اعمال را به‌صورت مرتب در یک پریود زمانی خاص انجام دهد (مثل تنظیم ساعت، چک کردن سخت‌افزار و...).
- **وقفه‌های I/O**: این وقفه‌ها به وسیله کنترل کننده‌های دستگاه I/O تولید می‌شوند تا کامل شدن طبیعی یک عمل یا بروز خطا در انجام عمل را نشان دهند.
- **وقفه‌های نقص سخت‌افزار یا وقفه‌های (Machine- check)**: مثل وقفه‌ای که بر اثر خطای بیت توازن (parity) حافظه رخ می‌دهد یا وقفه نقص برق.
- **وقفه Restart** که با فشار دادن دکمه Reset ایجاد می‌شود.
- **وقفه (Super Visor Call)**: که در واقع یک تقاضا از طرف برنامه کاربر جهت دریافت سرویس ویژه‌ای از سیستم‌عامل است. در یک تقسیم‌بندی کلی می‌توان وقفه‌ها را سه دسته کرد.

۱- وقفه‌های داخلی (trap) که بر اثر اجرای دستورات خود برنامه به‌صورت داخلی CPU رخ می‌دهند.

۲- وقفه‌های خارجی که از دستگاه‌های خارجی مثل دستگاه‌های ورودی یا خروجی در DMA تایمرها، صفحه کلید و خطاهای سخت‌افزاری ناشی می‌شوند.

۳- وقفه‌های نرم‌افزاری (یا همان SVC) که بر اثر فراخوانی توابع سیستمی توسط برنامه رخ می‌دهند.

با اجرای مجدد برنامه، وقفه‌های داخلی به همان صورت قبلی دوباره رخ می‌دهند ولی وقفه‌های خارجی مستقل از دستورات برنامه و ناهمگام با برنامه می‌باشند.

اگر چند منبع هم‌زمان تقاضایشان را از طریق یک خط وقفه به CPU اعلام کنند، آنگاه CPU با روش همه‌پرسی یا سرکشی (polling) منبع وقفه‌دهنده را تشخیص خواهد داد.

۲۵۶ سطح اولویت توسط پردازنده‌های ۸۶×۸۰ پشتیبانی می‌شود که می‌توان آن‌ها را به سه گروه کلی تقسیم کرد:

- وقفه‌های داخلی سخت‌افزاری
- وقفه‌های خارجی سخت‌افزاری
- وقفه‌های نرم‌افزاری

وقفه‌های داخلی سخت‌افزاری

وقفه‌های داخلی سخت‌افزاری (internal hardware- interrupts) به دلیل رخ دادن وضعیت معینی که در حین اجرای یک برنامه پیش آمده تولید می‌شوند (مانند تقسیم بر صفر).

وقفه‌هایی که بر اثر خطا به وجود می‌آید تله (trap) هم نامیده می‌شود تله باعث سقوط برنامه می‌شود.

این وقفه‌ها توسط سخت‌افزار اداره می‌شوند و امکان تغییر آن‌ها وجود ندارد. اما با وجودی که نمی‌توان آن‌ها را مستقیماً مدیریت کرد، این امکان وجود دارد که از اثر آن روی کامپیوتر به نحو مفیدی استفاده شود.

مثال: سخت‌افزار وقفه شمارنده ساعت کامپیوتر را چند بار در ثانیه فراخوانی می‌کند تا زمان را نگه دارد.

می‌توان برنامه‌ای نوشت که مقدار شمارنده ساعت را خوانده آن را به شکل قابل درک کاربر به‌صورت ساعت و دقیقه تبدیل کند.

وقفه‌های خارجی سخت‌افزاری

وقفه‌های خارجی سخت‌افزاری (external hard- interrupts) خارج از CPU و توسط دستگاه‌های جانبی، مانند صفحه کلید، چاپگر، کارت‌های ارتباطی و یا کمک پردازنده تولید می‌شوند.
دستگاه‌های جانبی با ارسال وقفه به CPU خواستار قطع اجرای برنامه فعلی شده و CPU را متوجه خود می‌کنند.
آن‌ها به پایه INTR (maskable interrupts) یا NMI (non maskable interrupts) پردازنده متصل هستند.
وقفه‌های دستگاه‌ها می‌توانند از طریق مداری به نام PIC 8259A، که کارش منحصراً سروکار داشتن با این نوع وقفه‌هاست، به پردازنده ارسال شوند.
مدار PIC (programmable interrupt controller) که توسط CPU کنترل می‌شود سیگنال‌هایش را روی پایه INTR قرار می‌دهد و امکان فعال و غیرفعال کردن وقفه‌ها و تغییر سطح اولویت را تحت نظارت یک برنامه می‌دهد.
دستورات STI و CLI می‌توانند برای فعال و غیر فعال کردن وقفه‌هایی که روی پایه INTR ارسال می‌شوند، بکار روند که البته روی وقفه‌هایی NMI تأثیری ندارد.

چند نمونه از وقفه‌های سخت‌افزاری:

IRQ_0: ساعت CPU
IRQ_1: صفحه کلید
IRQ_2: دومین کنترل‌کننده وقفه
IRQ_3: دومین درگاه سریال
IRQ_4: اولین درگاه سریال
IRQ_5: دومین درگاه موازی
IRQ_6: فلاپی درایو
IRQ_7: اولین درگاه موازی
IRQ_8: ساعت سیستم
IRQ_9: آزاد
IRQA: آزاد
IRQB: آزاد
IRQC: آزاد
IRQD: کمک پردازنده
IRQE: هارد دیسک
IRQF: آزاد

وقفه‌های نرم‌افزاری

وقفه‌های نرم‌افزاری (software interruptions) در نتیجه دستورالعمل int در یک برنامه در حال اجرا تولید می‌شوند. برنامه‌نویس می‌تواند با دادن دستور int یک وقفه نرم‌افزاری تولید کند. بدین طریق بلافاصله اجرای برنامه فعلی را متوقف می‌کند و CPU را به روتین وقفه هدایت می‌کند، برنامه‌نویس از طریق وقفه‌ها می‌تواند در برنامه با وسایل جانبی ارتباط برقرار کند. استفاده از وقفه‌ها باعث کوتاه‌تر شدن کد برنامه و درک آسان‌تر و اجرای بهتر آن می‌شود.

روتین‌های وقفه نرم‌افزاری بخشی از سیستم‌عامل هستند از این‌رو وقفه‌های نرم‌افزاری را می‌توان به دو گروه تقسیم کرد: وقفه‌های سیستم‌عامل DOS و وقفه‌های BIOS وقفه‌های DOS آسان‌تر استفاده می‌شوند اما از وقفه‌های BIOS که قسمتی از سخت‌افزار هستند کندتر هستند.

DOS این نوع وقفه‌ها را برای اجرای API (application programming interface) خودش استفاده می‌کند. بیشتر سیستم‌عامل‌های جدید مانند Unix Windows واسطه C-based را استفاده می‌کنند.

دستورالعمل int (interrupt): یک روتین وقفه را فراخوانی می‌کند، فرم کلی آن به‌صورت زیر است:

```
int n
```

n شماره وقفه موردنظر و مقداری بین ۰ تا ۲۵۵ است که اجازه فراخوانی ۲۵۶ روتین مختلف وقفه را می‌دهد.

دستورالعمل int یک فراخوانی سیستمی را می‌سازد و شکل خاصی از دستورالعمل فراخوانی یک زیربرنامه (دستورالعمل) call است.

مشکل دستورالعمل int این است که تنها ۲۵۶ روتین وقفه را می‌تواند پشتیبانی کند، در حالیکه DOS به تنهایی دارای بیش از ۱۰۰ سرویس مختلف وقفه و BIOS بیش از هزاران سرویس وقفه است که این تعداد بیش از کلیه وقفه‌هایی است که توسط اینتل رزرو شده است. برای حل این مشکل از یک شماره وقفه برای هر دسته از سرویس‌های وقفه و یک شماره تابع تعیین سرویس موردنظر استفاده می‌شود.

شماره تابع توسط یکی از ثبات‌ها (اکثراً AH) هنگام فراخوانی وقفه ارسال به روتین وقفه می‌شود.

مثال: سیستم‌عامل DOS شماره وقفه h۲۱ را بکار می‌گیرد برای انتخاب یک تابع خاص، قبل از فراخوانی وقفه، کد تابع در ثابت AH قرار می‌گیرد، برای نمونه تابع ۴Ch این وقفه برای خاتمه برنامه و برگشت به محیط DOS فراخوانی می‌شود.

```
mov AH , ۴Ch
Int ۲۱ h
```

جدول بردار وقفه

هر سطح وقفه یک محل رزرو شده در حافظه دارد که بردار وقفه (interrupt vector) نامیده می‌شود. همه بردارهای وقفه در جدولی به نام جدول برنامه وقفه (interrupt vector table) نگهداری می‌شوند، این جدول از ابتدای حافظه اصلی یعنی آدرس ۰۰۰۰:۰۰۰۰ ذخیره شده است.

هر بردار وقفه ۴ بایت طول دارد، دو بایت بالای آن آفست و دو بایت پایین آن سگمنت روتین وقفه را در برمی‌گیرند. چون ۲۵۶ روتین‌های وقفه وجود دارد، بنابراین اندازه جدول بردار وقفه ۱KB = ۱۰۲۴ = ۴×۲۵۶ است.

شماره وقفه به عنوان اندیسی برای جدول بردار وقفه استفاده می‌شود. آفست روتین وقفه شماره n در آدرس n×۴ و آدرس سگمنت روتین وقفه شماره n در آدرس n×۴+۲ جدول قرار دارد.

یک ویژگی خوب این سیستم این است که می‌توان بردارها را برای اشاره به روتین دیگری تغییر داد، که این همان کاری است که برنامه‌های TSR (Terminate and Stay Resident) انجام می‌دهند.

در برنامه همیشه توسط شمار وقفه به یک روتین وقفه مراجعه می‌شود بنابراین برنامه نیازی به دانستن آدرس واقعی در حافظه ندارد و آدرس روتین وقفه هنگام اجرا توسط CPU تعیین می‌شود.

مثال: آدرس‌های آفست و سگمنت روتین وقفه شماره ۵ برابر با ۵×۴ = ۰۰۱۴h و ۵×۴+۲ = h۰۰۱۶ می‌باشد.

چند نمونه وقفه متعارف

وقفه ۲۱h

تابع ۰۱h یک کلید را از صفحه کلید می‌خواند، کد کلید خوانده شده در ثبات AL برگردانده می‌شود.

```
mov AH , ۴Ch
Int ۲۱ h
mov AL , character
```

تابع ۰۲ h یک کاراکتر را روی صفحه نمایش نشان می‌دهد که کد کاراکتر در ثبات DL باید قرار داده شود.

```
mov AH , ۰۲h
mov DL , character
Int ۲۱ h
```

تابع ۰۹ h یک رشته کاراکتری را روی صفحه، نمایش می‌دهد. آدرس شروع رشته باید در ثبات‌های DS:DX قرار بگیرد. انتهای رشته توسط کاراکتر ($) باید تعیین شده باشد.

```
mov  AH,۹h
Lea DX, string
Int ۲۱ h
```

وقفه ۱۰ h

تابع ۰۲h مکان‌نما را روی صفحه نمایش به سطر و ستون خاصی منتقل می‌کند. شماره صفحه در BH، شماره سطر در DH و شماره ستون در DL باید قرار بگیرد. مختصات صفحه از نقطه (۰,۰) از گوشه بالای چپ صفحه نمایش شروع می‌شود.

```
mov AH , ۰۲h
mov BH , page
mov DH , row
mov DL  , coulmn
Int ۱۰h
```

تابع ۰۹h یک کاراکتر را با رنگ معین چندبار نمایش می‌دهد. کد کاراکتر در AL، شماره صفحه در BH، خاصیت رنگ کاراکتر در BL و تعداد کاراکتر در CX باید قرار بگیرد.

```
mov AH , ۰۹h
mov AL  , character
mov BH , page
mov BL , attribute
mov CX, number
Int ۱۰h
```

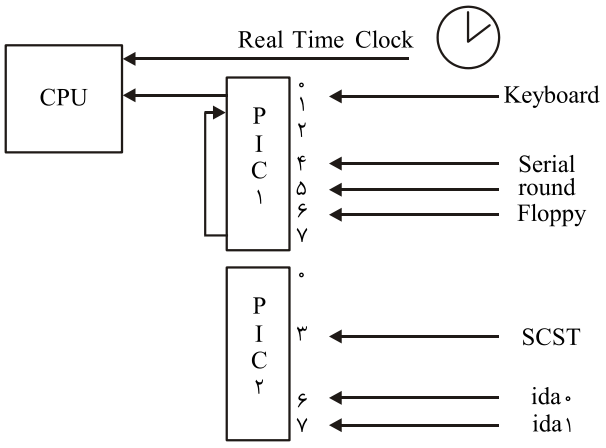


Figure: A Logical Diagram of Intrrupt Routing

رایج‌ترین وقفه‌ها عبارتند از:

۱- برنامه: وقفه‌ای است که در برخی شرایط خاص یک دستورالعمل از جمله سرریز و تقسیم به صفحه رخ می‌دهد.

۲- زمان‌سنج: وقفه‌ای است که توسط زمان‌سنج داخلی پردازنده تولید می‌شود.

۳- ورودی خروجی: این وقفه به وسیله کنترل‌کننده ورودی/ خروجی ایجاد می‌شود.

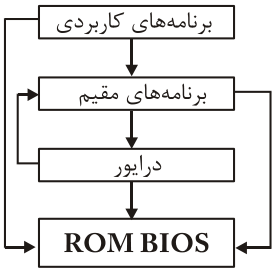
۴- نقص سخت‌افزار: این وقفه توسط سخت‌افزار تولید می‌شود.

سیستم‌عامل از یک سری روتین تشکیل شده است. اینکه این روتین‌ها چگونه با هم در ارتباط هستند نوع سیستم عامل را تعیین می‌کند و ساختارهای مختلف سیستم‌عامل را به‌وجود می‌آورد.

ساختارهای مختلف سیستم‌عامل

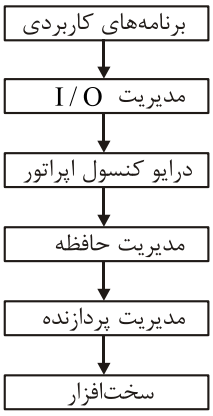
۱ – ساختار ساده (Monolithic)

ساده‌ترین نوع است. در اینجا سیستم‌عامل مانند کیسه‌ای است که تمام روتین‌ها در داخل آن قرار دارد و همه روتین‌ها همدیگر را می‌شناسند و می‌توانند همدیگر را فراخوانی کنند. از جمله این سیستم‌عامل‌ها، DOS و Unixهای اولیه بودند. حفاظت در این نوع سیستم‌عامل‌ها بسیار ضعیف بود.



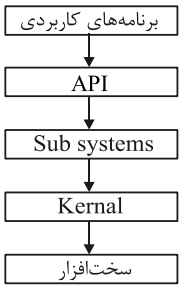
۲– لایه‌بندی (Layered)

پروفسور دایکسترا اولین سیستم‌عامل لایه‌بندی شده را به نام THE (Techinische Hoghschoole Eindhover) ارائه کرد که ساختار آن در زیر آمده است:



در این نوع از سیستم‌عامل‌ها روتین‌ها دسته‌بندی می‌شوند و هر دسته یک لایه را تشکیل می‌دهند. بالاترین لایه روتین‌های کاربر و پایین‌ترین لایه روتین‌های سخت‌افزار می‌باشد. در این سیستم لایه بالایی می‌تواند لایه زیرین بلا فصل خود را فراخوانی کند و از چگونگی انجام عمل در لایه زیرین بی اطلاع است. پس از انجام کار توسط لایه زیرین، نتیجه به لایه بالایی تحویل می‌شود.

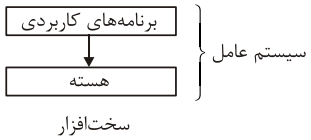
نکته: نسخه‌ای از DOS به نام OS/۲ از سیستم لایه‌بندی استفاده می‌کرد که ساختار آن به‌صورت زیر می‌باشد:



مزیت سیستم‌های لایه‌بندی شده، حفاظت قوی آنها بود ولی عیب آنها، طراحی مشکلشان بود. ویندوزهای اولیه سیستم لایه‌بندی استفاده می‌کردند.

۳- ساختار ماشین مجازی (Virtual machine)

در این روش سیستم‌عامل از دو لایه تشکیل شده است که به‌صورت زیر می‌باشد:



در اینجا یک یا چند سیستم‌عامل می‌تواند بر روی لایه برنامه‌های سیستم مستقر شود. از جمله این سیستم‌عامل‌ها برای ماشین‌های پنتیوم سیستم‌عامل Exo kernel می‌باشد که ساختار ماشین مجازی دارد. این سیستم‌عامل، سیستم را به دو یا سه ماشین مجازی تبدیل می‌کند مثلاً روی Exokernel می‌توانند سیستم‌عامل‌های DOS، Windows، OS/2 قرار بگیرند.

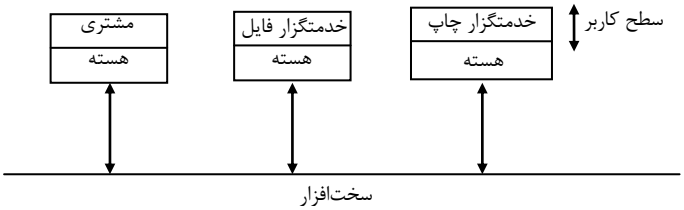
مزیت این سیستم‌ها این است که حفاظت بین کاربران زیاد است ولی از معایب آن حجم بسیار بالای آن می‌باشد بنابراین روی خیلی از ماشین‌ها قابل استفاده نیستند.

۴- ساختار Microkernel

در این ساختار روتین‌ها را به بخش‌هایی تقسیم می‌کنند که هر یک مسئول انجام کار خاصی می‌باشند. آنهایی که باید با سخت‌افزار کار کنند را در لایه‌های زیرین و بقیه را در لایه‌های بالایی قرار می‌دهند. ساختار آن مانند شکل زیر می‌باشد:

۵- ساختار خدمتگزار - مشتری (Client – Server)

از این ساختار در سیستم‌های توزیع شده استفاده می‌شود و در آن به‌عنوان مثال یک کامپیوتر مسئول کارکردن با فایل‌ها و کامپیوتر دیگر مسئول چاپ و... می‌باشد. البته همه این هسته‌ها نیز می‌توانند در یک کامپیوتر باشند. ساختار آن به‌صورت شکل زیر است:



سوالات چهارگزینه‌ای سراسری فصل اول

۱- از بخش‌های اصلی یک نرم‌افزار در اداره‌کننده دستگاه (Device driver) می‌توان موارد زیر را برشمرد. **(مهندسی کامپیوتر ۷۹)**

۱) هسته سیستم‌عامل، زمان‌بند ورودی/خروجی، روتین‌های خدماتی و وقفه‌های این دستگاه
۲) مدیریت دستگاه، زمان‌بندی فرآیندها و زمان‌بندی ورودی/خروجی
۳) تبدیل‌کننده استانداردهای ورودی/خروجی به یکدیگر، روتین خدماتی وقفه، برنامه‌های ورودی/خروجی و هدایت دستگاه
۴) نرم‌افزارهای میانگیرگیری(Buffering) ، مدیریت پرونده ها، زمان‌بندی ورودی/خروجی و حفاظت

۲- کدام‌یک از موارد زیر در یک سیستم real time درست است؟ **(فناوری اطلاعات ۸۵)**

۱) تنها ملاک درستی انجام یک 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 آن که بتوان به کارهای با اولویت بالاتر پاسخ داد، یک پردازنده نمی‌تواند مدت زمان زیادی در kernel باشد.
۴) اگر یک کار deadline نداشته باشد، ممکن است هیچ‌گاه CPU را در اختیار نگیرد یعنی گرسنگی حاصل شود.

۳- مدیریت منابع زیر با کدام عامل است؟ **(فناوری اطلاعات ۸۶)**

۱- ثبات‌ها، ۲-حافظه پنهان (cache)، ۳- حافظه اصلی،۴-فضای دیسک

۱) ۱- کاربر ۲- سیستم‌عامل ۳- سیستم‌عامل ۴- سیستم‌عامل
۲) ۱- کامپایلر ۲-خودکاراست (سخت‌افزاری)۳-سیستم‌عامل۴- سیستم‌عامل
۳) ۱- کاربر ۲- سیستم‌عامل ۳- کامپایلر ۴- خودکار است(سخت‌افزاری)
۴) ۱- خودکاراست (سخت‌افزاری)۲- خودکاراست (سخت‌افزاری) ۳- سیستم‌عامل ۴- سیستم‌عامل یا خودکار

۴- فرض کنید دو فرآیند P_1 و P_2 روی یک سیستم Multi- task قرار داشته باشند. اگر P_1 وظیفه مرتب‌کردن لیستی بزرگ از داده‌ها با طول دلخواه را برعهده داشته باشد و P_2 یک عملیات محاسباتی دیگر روی هرکدام از داده‌های لیست مرتب شده انجام دهد، کدام روش برای تبادل اطلاعات بین این دو فرآیند مناسب‌تر است؟ **(فناوری اطلاعات ۸۸)**

۱) ارتباط از طریق pipe
۲) حافظه مشترک (shared memory)
۳) تبادل پیام (massage passing)
۴) مدل سرویس‌گیر- سرویس‌گر (client- server)

۵- فرض کنید زمان محاسبات یک فرآیند ۲۰۰سیکل cpu باشد، در ضمن عملیات I/O در حال انجام یک فرآیند دیگر از طریق DMA بوده است و پس از ۱۰۰سیکل cpu پایان عملیات I/O توسط یک وقفه به سیستم اطلاع داده می‌شود. اگر زمان اجرای IRS را ۱۰ سیکل cpu فرض کنیم، کل عملیات مذکور چه مقدار زمان سیستم را به خود اختصاص می‌دهند؟ (زمان هر سیکل cpu معادل یک واحد زمانی فرض می‌شود.) **(فناوری اطلاعات ۸۸)**

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

۶- کدام‌یک از جملات زیر غلط است؟ **(علوم کامپیوتر ۸۸)**

۱) در حالت کلی، مفهوم عمل وقفه و polling یکسان است ولی سرعت وقفه بیش‌تر است.
۲) روش polling در زمان‌هایی که تعداد خطوط وقفه پردازنده کم است بیش‌تر مورد استفاده قرار می‌گیرد.
۳) روش polling، روشی برای بررسی درخواست‌های دستگاه‌های جانبی توسط پردازنده است.
۴) عملکرد روش وقفه نسبت به polling بهتر است.

۷- هدف اصلی از عملیات dual – mode چیست؟ **(فناوری اطلاعات ۹۲)**

۱) تشخیص خطاها
۲) کاهش توان مصرفی کامپیوتر
۳) توانایی سیستم‌عامل در کنترل پردازنده
۴) محافظت سیستم‌عامل از دیگر نرم‌افزارها

پاسخنامه سوالات چهارگزینه‌ای سراسری فصل اول

۱- گزینه «۳»

زمان‌بند ورودی و خروجی از وظایف سیستم‌عامل است. گزینه ۱ به علت عبارت "هسته سیستم‌عامل" غلط است.

۲- گزینه «۴»

البته گزینه ۲ هم درست است چرا که به دلیل نیاز به پاسخ‌دهی سریع و تضمین شده، سیستم‌های بلادرنگ از حافظه مجازی و اشتراک زمانی استفاده نمی‌کنند.

۳- گزینه «۲»

مدیریت حافظه اصلی و فضای دیسک بر عهده سیستم‌عامل است حافظه نهان نیز به‌وسیله مدیر حافظه نهان به‌صورت سخت‌افزاری مدیریت می‌شود و ثبات‌ها هم به‌وسیله کدی که کامپایلر تولید می‌کند کنترل می‌شوند.

۴- گزینه «۳»

به‌طور کلی بهترین روش برای تبادل اطلاعات به‌منظور هماهنگی دو فرآیند، تبادل پیام است اما ممکن است منظور سوال، لیست مشترک بین دو فرآیند باشد که در این‌صورت، چون این لیست بزرگ است، بهترین روش استفاده از حافظه مشترک است.

۵- گزینه «۴»

از آنجا که عمل I/O و محاسبات به‌صورت هم‌زمان انجام می‌شوند پس این دو کار ۲۰۰ سیکل طول می‌کشند. برای پردازش وقفه هم ۱۰ سیکل نیاز است که جمعا ۲۱۰ می‌شود اما از آنجا که زمانی هم برای تعویض متن در هنگام بروز وقفه نیاز است بنابراین بیشتر از ۲۱۰ سیکل طول می‌کشد.

۶- گزینه «۲»

از آنجا که وقتی چند منبع به‌صورت هم‌زمان و از طریق یک خط وقفه، تقاضایشان را به CPU می‌فرستند. CPU با استفاده از تکنیک Polling منبع وقفه‌دهنده را شناسایی کرده و سرویس می‌دهد بنابراین تعداد خطوط وقفه بیشتر برای روش Polling موردنیاز است.

۷- گزینه «۳»

منظور از سیستم dual mode سیستمی است که دارای دو مود کاربر و هسته می‌باشد که به این طریق فضای هسته از فضای کاربر جدا می‌شود و بدین‌ترتیب سیستم‌عامل از نرم‌افزارهای سطح کاربر محافظت می‌شود.

سوالات چهارگزینه‌ای آزاد فصل اول

۱- در محیط یک سیستم‌عامل چند وظیفه‌ای (multi-task) کدام‌یک از وقفه‌های زیر از اولویت بالاتری برخوردار می‌باشند؟

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

۲- کدام‌یک از عملیات زیر از تله (Trap) سیستم‌عامل استفاده نمی‌کند؟ **(مهندسی کامپیوتر۸۲)**

۱) فراخوانی سیستم ۲) دسترسی غیرمجاز به حافظه ۳) مقداردهی متغیرها ۴) نقص صفحه

۳- کدام نوع سیستم‌عامل امکان ارائه ماشین مجازی محاوره‌ای را برای کاربران فراهم می‌نماید؟ **(مهندسی کامپیوتر۸۲)**

۱) دسته‌ای ۲) کامپیوتر شخصی ۳) بلادرنگ ۴) اشتراک زمانی

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

(فناوری اطلاعات ۸۳)

۱) این ساختار در مقابل رخداد خطا قابلیت اطمینان پایین دارد.
۲) این ساختار برای سیستم توزیع شده مناسب است.
۳) این ساختار به‌صورت لایه عمودی طراحی شده است.
۴) این ساختار مبتنی بر کم‌تر نمودن کدهای موجود در هسته (kernel) است.

۵- کدام گزینه از مزایای سیستم‌عامل ماشین مجازی نیست؟ **(فناوری اطلاعات ۸۶)**

۱) سرعت بالا ۲) افزایش امنیت برای سیستم
۳) حل مشکلات ناسازگاری سیستم ۴) استفاده چندین کاربر از یک سیستم

۶- کدام گزینه از وظایف سیستم‌عامل در رابطه با مدیریت سیستم I/O نیست؟ **(فناوری اطلاعات ۸۶)**

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

۷- کدام گزینه از وظایف سیستم‌عامل در رابطه با مدیریت فرآیند است؟ **(فناوری اطلاعات ۸۶)**

۱) مدیریت بن‌بست ۲) مدیریت حافظه پر/خالی برای فرآیندها
۳) مدیریت زمان‌بندی دیسک ۴) مدیریت اسپولینگ

۸- امکان انتقال یک کلمه توسط واحد DMA در کدام نقطه از یک چرخه دستورالعمل وجود ندارد؟ **(فناوری اطلاعات ۸۸)**

۱) پس از واکشی عملونده ۲) پس از کدگشایی دستورالعمل
۳) پس از ذخیره نتایج ۴) پس از اجرای دستورالعمل

پاسخنامه سوالات چهارگزینه‌ای آزاد فصل اول

۱- گزینه «۳»

۲- گزینه «۳»

مقداردهی متغیرها جز وقفه‌های نرم‌افزاری محسوب نمی‌شود.

۳- گزینه «۴»

در سیستم‌های اشتراک زمانی، هریک از کاربران می‌توانند به‌صورت مجازی یک کامپیوتر را در اختیار داشته باشند.

۴- گزینه «۱»

به علت اجرای کلیه پروسس‌های خدمتگزار در مد کاربر هیچ کدام از آنها دسترسی مستقیم به سخت‌افزار ندارند.

۵- گزینه «۱»

به‌خاطر وجود ماشین مجازی جاوا، برنامه‌های جاوا، کندتر از برنامه‌های زبان C هستند.

ماشین‌های مجازی سرعت کمتری دارند.

۶- گزینه «۳»

مدیریت بافرینگ بیشتر مربوط به مدیریت حافظه است.

۷- گزینه «۱»

گزینه ۲ مربوط به مدیر حافظه و گزینه ۳ مربوط به مدیریت وسایل I/O است.

۸- گزینه «۳»

پس از واکشی عملوندها CPU محاسبات را انجام می‌دهد لذا DMA می‌تواند از خطوط استفاده کند، پس از کدگشایی دستور هم ممکن است اجرای دستور نیاز به خواندن یا نوشتن حافظه نداشته باشد. پس از اجرای کامل یک دستور DMA می‌تواند از خطوط استفاده کند.

تنظیم ساعت سیستم مهم می‌باشد بنابراین می‌بایست در مد هسته و تحت نظارت سوپروایزر باشد.

فصل دوم

فرآیندها و نخ‌ها

- حالات اجرا
- بلوک کنترلی فرآیند
- تعویض متن
- مفهوم نخ و تکنیک چندنخی

فرآیندها و نخ‌ها

فرآیند: برنامه در حال اجرا را فرآیند گویند. پس فرآیند حالت فعال ولی برنامه حالت غیر فعال دارد. بعد از آنکه به برنامه‌ای پردازنده تخصیص داده می‌شود، برنامه به فرآیند تبدیل می‌شود.

هر فرآیند شامل موارد زیر است:

- شمارنده برنامه
- پشته
- بخش داده‌ها

هر فرآیند در هنگام اجرا حالت خود را تغییر می‌دهد. رفتار یک فرآیند بخصوص را می‌توان با دنبال کردن **رد** آن فرآیند مشخص نمود. منظور از رد یک فرآیند، فهرستی از دنباله دستورالعمل‌هایی است که برای آن فرآیند اجرا می‌شوند. بنابراین با نمایش چگونگی تداخل ردهای فرآیندهای مختلف، می‌توان رفتار پردازنده را مشخص کرد.

ایجاد و پایان فرآیند

عمر هر فرآیندی محدود به زمان ایجاد تا زمان پایان دادن به آن فرآیند است. هنگام اضافه شدن یک فرآیند جدید به فرآیندهای تحت مدیریت سیستم‌عامل، ساختمان داده مورد نیاز آن فرآیند توسط سیستم‌عامل ساخته می‌شود و همچنین فضای آدرس مورد استفاده آن در حافظه اصلی به آن اختصاص می‌یابد که این کار نیز توسط سیستم‌عامل انجام می‌شود. به‌طور معمول به چهار دلیل یک فرآیند ایجاد می‌شود:

۱- یک فرآیند جدید بر اساس درخواست صریح یک فرآیند دیگر، توسط سیستم‌عامل ایجاد می‌گردد که به این عمل زایش فرآیند می‌گویند. فرآیند ایجاد شده، **فرزند** و فرآیند درخواست دهنده، **پدر** نامیده می‌شود.

۲- در محیط دسته‌ای، یک فرآیند جدید در پاسخ به تحویل یک کار، ساخته می‌شود.

۳- در محیط محاوره‌ای، هنگامی یک کاربر جدید از طریق یک پایانه اقدام به برقراری ارتباط با کامپیوتر می‌کند در واقع یک فرآیند جدید ایجاد می‌شود.

۴- سیستم‌عامل می‌تواند یک فرآیند جدید را از طرف یک برنامه کاربردی برای ارائه خدمتی ایجاد کند. مثلا در هنگام تقاضای چاپ یک پرونده، سیستم‌عامل می‌تواند یک فرآیند جدید برای مدیریت آن چاپ ایجاد کند.

به دلایل زیر ممکن است یک فرآیند پایان یابد:

۱- اگر یک فرآیند به مقدار حافظه‌ای بیش از آنچه که سیستم می‌تواند برای او فراهم کند نیاز داشته باشد فرآیند پایان خواهد یافت.

۲- ممکن است خطایی در ورودی/خروجی اتفاق افتاده باشد و موجب پایان یافتن فرآیند شود مثل انجام یک عمل نامعتبر همچون خازن روی چاپگر.