

فصل ۱:

مفاهیم اولیه

سیستم عامل از منابع سخت افزاری پردازنده برای ارائه خدمات به کاربران استفاده می کند. بنابراین آشنایی با سخت افزار کامپیوتر، برای بررسی سیستم عامل ضروری است. اجزای سخت افزاری تشکیل دهنده کامپیوتر عبارتند از:

- ۱- پردازنده
- ۲- حافظه اصلی
- ۳- مولفه های ورودی و خروجی
- ۴- اتصالات داخلی سیستم.

پردازنده

پردازنده از واحد محاسبه و منطق (ALU)، واحد کنترل و رجیسترها (ثبات ها) تشکیل می شود و سه گام "واکشی، رمز گشایی و اجرا" را به طور مداوم انجام می دهد. پردازنده دائما در حال کار است و فقط در موارد خاصی به مدت کوتاهی به حالت Hold می رود و کنترل گذرگاه را به کنترل کننده DMA می دهد.

ثباتهای پردازنده

در داخل پردازنده مجموعه ای از ثباتها وجود دارد. این ثباتها سطحی از حافظه که سریعتر و کوچکتر از حافظه اصلی است را فراهم می کنند. ثباتهای داخل پردازنده عبارتند از:

۱- ثباتهای داده (AX, BX, CX, DX)

۲- ثبات شاخص (SI, DI)

۳- اشاره گر قطعه (CS, DS, SS, ES)

۴- اشاره گر پشته (SP)

۵- شمارنده برنامه (PC یا IP)

۶- ثبات دستورالعمل (IR)

تذکر: تمام پردازنده ها شامل یک یا مجموعه ای از ثباتها هستند به نام کلمه وضعیت (PSW) که حاوی اطلاعات وضعیت هستند. این ثبات، نوعا علاوه بر کدهای وضعیت، شامل اطلاعات دیگری، مثل بیت فعال/غیر فعال کردن وقفه و بیت حالت کاربر/سرپرست، نیز می باشد.

حالت‌های اجرای پردازنده

پردازنده دارای دو حالت اجرا (dual-mode) می باشد:

- ۱- **مد کاربر:** حالت کم امتیازتری که برنامه های کاربران در این حالت اجرا می شوند.
- ۲- **مد هسته (سیستم، کنترل، کرنل):** حالت ممتازتری که دستورالعملهایی مانند تغییر ثباتهای کنترل، I/O و مدیریت حافظه، در این حالت اجرا می شوند. در این حالت نرم افزار کنترل کامل پردازنده، دستورالعملها، ثباتها و حافظه را در اختیار دارد. این سطح از کنترل برای برنامه های کاربران غیرضروری، ناامن و نامطلوب است.

هدف اصلی از عملیات dual-mode، محافظت سیستم عامل از دیگر نرم افزارها می باشد.

فرایند یا پردازش (process)، یک برنامه در حال اجرا می باشد. تفاوت برنامه و فرایند در این است که برنامه یک نهاد غیرفعال و فرایند یک نهاد فعال می باشد. معمولاً برنامه بر روی دیسک به صورت یک فایل اجرایی دودویی ذخیره می شود. برنامه باید به حافظه لود شود و در داخل فرایندی قرار بگیرد تا اجرا شود. فرایند می تواند هم در حافظه اصلی و هم در حافظه جانبی قرار بگیرد.

تنظیم زمان سیستم در مد کرنل و خواندن ساعت سیستم در مد کاربر انجام می شود.

پردازنده به کمک بیتی که در PSW وجود دارد متوجه می شود که باید در کدام حالت (کاربر یا هسته) اجرا کند.

وقتی کاری برای اجرا موجود نباشد، پردازنده یک برنامه شامل یک حلقه انتظار مشغول را اجرا خواهد کرد.

عمل تغییر وضعیت از مد کرنل به مد کاربر و بالعکس از امکانات سخت افزاری است که برای کمک مستقیم به سیستم عامل طراحی شده است.

وقفه (interrupt)

وقفه سیگنالی است که روند عادی اجرا را تغییر می دهد. وقتی وقفه‌ای به CPU ارسال می شود، کار CPU متوقف شده و روال پاسخگو به وقفه (interrupt routine) اجرا می شود. بعد از پایان اجرای این روتین، CPU کار قبلی اش را ادامه می دهد. آدرس دستوری که هنگام اجرای آن وقفه صادر شده (آدرس برگشت) در پشته ذخیره می شود. بعد از پایان پاسخگویی به وقفه، آدرس برگشت در شمارنده برنامه (PC) قرار می گیرد و محاسباتی که اجرای آنها به تعویق افتاده از سر گرفته می شود.

- با وجود وقفه‌ها، پردازنده می‌تواند در هنگامی که عمل ورودی/خروجی در جریان است، مشغول اجرای دستور عملهای دیگر باشد.
- سخت افزار در هر زمان می‌تواند با ارسال سیگنالی به CPU، وقفه‌ای را صادر کند. نرم افزار نیز با اجرای عملیات خاصی به نام فراخوانی سیستم (System call)، می‌تواند وقفه‌ای را صادر کند.
- وقتی وقفه‌ای رخ می‌دهد، سخت افزار کنترل را به سیستم عامل بر گردانده و ثبات‌ها و شمارنده برنامه را ذخیره می‌کند. همچنین نوع وقفه را نیز تعیین می‌کند.
- بردار وقفه، جدولی از اشاره گرها می‌باشد که هر اشاره گر به یک روال وقفه اشاره می‌کند.

انواع وقفه‌ها

وقفه‌ها بر چهار نوع می‌باشند:

۱- برنامه

وقفه‌هایی که به دلیل بعضی شرایط حاصل از اجرای یک دستورالعمل بروز می‌کند:

- الف- سرریز شدن محاسباتی
- ب- تقسیم بر صفر
- ج- تلاش برای اجرای یک دستورالعمل ماشین غیرمجاز
- د- مراجعه به آدرس خارج از فضای مجاز کاربر

۲- زمان سنج

وقفه‌ای که توسط زمان سنج داخلی پردازنده تولید می‌شود. این وقفه به سیستم عامل اجازه می‌دهد، بعضی اعمال (مانند تست حافظه، چک کردن سخت افزار و یا تعیین زمان اجرای پردازنده در هر برش در سیستم استراک زمانی) را به طور مرتب انجام دهد.

۳- ورودی/خروجی

وقفه‌هایی که به وسیله کنترل کننده I/O تولید می‌شود، تا کامل شدن طبیعی یک عمل یا شرایط خطا را اعمال کند.

۴- نقص سخت افزار

وقفه‌هایی که با نقص سخت افزاری تولید می‌شود، مثل نقص برق یا خطای توازن حافظه.

پردازش وقفه

هنگامی که یک دستگاه I/O، یک عمل I/O را کامل می‌کند، دنباله حوادث زیر اتفاق می‌افتد:

- ۱- دستگاه یک علامت وقفه برای پردازنده می‌فرستد.
- ۲- پردازنده اجرای دستورالعمل جاری را قبل از پاسخ به این وقفه، به پایان می‌رساند.
- ۳- پردازنده بروز وقفه را بررسی کرده، در می‌یابد که وقفه ای آمده است و علامتی مبنی بر دریافت وقفه برای دستگاه وقفه دهنده می‌فرستد.
- ۴- پردازنده برای انتقال کنترل به روال خدماتی مربوط آماده می‌شود. در ابتدا اطلاعات مورد نیاز برای از سرگیری برنامه جاری را ذخیره می‌کند. حداقل اطلاعات، محتویات ثبات وضعیت برنامه (PSW) و محل دستورالعمل بعدی (PC) می‌باشد. این اطلاعات می‌تواند در بالای پشته کنترل سیستم گذاشته شود.
- ۵- پردازنده شمارنده برنامه (PC) را با آدرس شروع برنامه گرداننده وقفه ای که قرار است به این وقفه پاسخ دهد، بار می‌کند. در این حالت پردازنده به چرخه واکنشی رفته و کنترل به برنامه گرداننده وقفه منتقل می‌شود.
- ۶- در این لحظه محتوای PC و PSW مربوط به برنامه وقفه داده شده، در پشته سیستم ذخیره است.
- ۷- روال وقفه را پردازش می‌کند.
- ۸- بعد از کامل شدن پردازش وقفه، مقادیر ثباتها از پشته POP شده و در ثباتها گذاشته می‌شود.
- ۹- بار کردن مجدد مقادیر PSW و PC از پشته، که موجب اجرای دستورالعمل بعدی از برنامه وقفه داده شده می‌شود.

تعویض متن (Context Switch)

هنگامیکه وقفه ای رخ می‌دهد، قبل از اینکه سیستم عامل کنترل را به یک روال وقفه گیر بخصوص رد کند، وضعیت پردازش جاری را در محلی حفظ می‌کند، تا بتواند بعداً آنرا ادامه دهد و سپس به طرف روال وقفه گیر می‌رود که به این جریان تعویض متن می‌گویند.

فراخوانی‌های سیستم (System Calls)

فراخوانی‌های سیستم واسطی بین فرایند و سیستم عامل فراهم می‌کند که به صورت دستورات زبان اسمبلی می‌باشند. در بعضی از سیستم‌ها، فراخوانی‌ها مستقیماً از برنامه‌های زبان سطح بالا ساخته شده و مانند زیر برنامه می‌باشند. وقتی برنامه ای اجرا می‌شود از فراخوانی‌های سیستمی به وفور استفاده می‌کند و کاربر جزئیات آن را نمی‌بیند.

انواع فراخوانی های سیستم عبارتند از:

- ۱- کنترل فرایند
- ۲- دستکاری فایلها
- ۳- دستکاری دستگاه
- ۴- دستکاری اطلاعات
- ۵- ارتباطات

وقتی یک وقفه، تله و یا فراخوانی سرپرست رخ می دهد، پردازنده در حالت هسته گذاشته شده و کنترل به سیستم عامل منتقل می گردد و یک تعویض حالت به یک روال سیستم عامل انجام شده، ولی چون اجرا در داخل فرایند جاری ادامه می یابد، تعویض فرایند انجام نمی شود.

حفاظت

مسئله حفاظت از سه دیدگاه مورد بررسی است:

۱- حفاظت از I/O

برای حفاظت از I/O می توان تمام دستورات I/O را به عنوان دستورات ممتاز در نظر گرفت تا کاربران فقط از طریق سیستم عامل بتوانند آن دستورات را اجرا کنند.

۲- حفاظت از حافظه

حفاظت حافظه را حداقل برای بردار وقفه و روال وقفه باید فراهم کرد. در واقع می خواهیم سیستم عامل را از دستیابی برنامه کاربر و همچنین برنامه های کاربر را از یکدیگر محافظت کنیم. یکی از روشهای ممکن استفاده از ثباتهای پایه و حد می باشد.

۳- حفاظت از CPU

باید کاری کرد که برنامه کاربر در حلقه گیر نکند و کنترل را به سیستم عامل برگرداند. برای این منظور از یک تایمر استفاده می کنیم.

سلسله مراتب حافظه

بین سه ویژگی کلیدی حافظه، یعنی "هزینه، ظرفیت و زمان دسترسی" باید سبک و سنگین کرد. برای این کار نمی توان بر یک حافظه یا فن آوری خاصی تکیه کرد و باید از سلسله مراتب حافظه استفاده کرد. یک سلسله مراتب متداول حافظه در زیر آورده شده است:

- ۱- ثباتها ۲- حافظه پنهان ۳- حافظه اصلی
- ۴- حافظه پنهان دیسک ۵- دیسک مغناطیسی ۶- رسانه جا به جایی پذیر

با حرکت به سطوح پایین تر این سلسله مراتب، شرایط زیر رخ می دهد:

- ۱- کاهش هزینه در هر بیت ۲- افزایش ظرفیت
 - ۳- افزایش زمان دسترسی ۴- کاهش تعداد دفعات دسترسی پردازنده به حافظه
- مدیریت منابع هر یک از حافظه های زیر در مقابل آن نوشته شده است:

ثبات: کامپایلر

حافظه پنهان : خودکار(سخت افزاری)

حافظه اصلی و دیسک: سیستم عامل

حافظه پنهان

حافظه پنهان (cache)، یک حافظه کوچک و سریع بین پردازنده و حافظه اصلی است. حافظه پنهان، حاوی بخشی از حافظه اصلی است. وقتی پردازنده می خواهد کلمه ای از حافظه را بخواند، وجود آن را در حافظه پنهان بررسی می کند. اگر وجود داشته باشد، به پردازنده تحویل می شود، در غیر اینصورت یک بلوک از حافظه اصلی، شامل تعداد ثابتی از خانه های حافظه، به حافظه پنهان منتقل شده و سپس کلمه مورد نظر به پردازنده تحویل داده می شود. هنگامی که یک بلوک از داده ها به حافظه پنهان آورده می شود تا یک مراجعه به حافظه انجام شود، به دلیل پدیده محلی بودن مراجعات، احتمالاً بزودی به دیگر کلمات آن بلوک نیز مراجعه خواهد شد.

روشهای انتقال ورودی/خروجی

برای عملیات I/O روشهای زیر وجود دارد:

۱- I/O برنامه سازی شده (سرکشی)

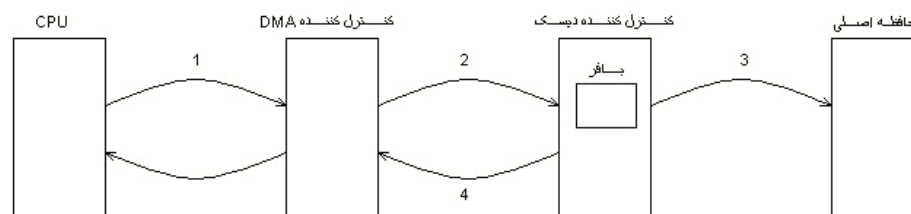
پردازنده یک فرمان I/O را از جانب فرایندی به یک مولفه I/O صادر می کند. سپس آن فرایند قبل از ادامه، تا کامل شدن عمل I/O، به انتظار مشغولی می گذراند.

۲- I/O مبتنی بر وقفه

پردازنده یک فرمان I/O را از جانب فرایند صادر می کند، سپس به اجرای دستورالعملهای بعدی ادامه می دهد و با کامل شدن عمل I/O، با وقفه مولفه I/O مواجه می شود.

۳- دسترسی مستقیم به حافظه (DMA)

مولفه DMA تبادل داده‌ها بین حافظه اصلی و مولفه I/O را کنترل می‌کند. پردازنده تقاضایی برای انتقال یک بلوک از داده‌ها را به مولفه DMA می‌فرستد و فقط پس از کنترل کل بلوک، مورد وقفه قرار می‌گیرد.



تذکر: در روش ورودی/خروجی مبتنی بر وقفه، دستورالعمل‌های بعدی می‌توانند از همان فرایند باشند، البته به شرطی که فرایند نیازی به انتظار برای تکمیل I/O نداشته باشد. در غیر اینصورت فرایند در انتظار وقفه معلق می‌گردد و کار دیگری انجام می‌گیرد.

روش دوم کارآمدتر از روش اول است.

برای انتقال I/O چند کلمه ای، روش DMA بسیار کارآمدتر از روشهای اول و دوم است.

در روش اول، مولفه I/O به پردازنده وقفه نمی‌دهد و این مسئولیت پردازنده است که وضعیت مولفه I/O را متناوباً بررسی کند تا اتمام آن عمل را دریابد. بنابراین روش اول را می‌توان ورودی/خروجی بر اساس سرکشی (polling) نامید.

مشکل ورودی/خروجی برنامه سازی شده این است که پردازنده باید برای مدت طولانی منتظر بماند تا مولفه I/O برای دریافت یا ارسال آماده شود. در مدت انتظار، پردازنده به علت اینکه مکرراً وضعیت را سؤال می‌کند، کارایی سیستم به شدت پایین می‌آید.

روش Programed I/O و Interrupt I/O پردازنده اصلی را درگیر عملیات I/O می‌کند.

نگاهی کلی به سیستم عامل

سیستم عامل برنامه ای است که اجرای برنامه های کاربردی را کنترل و به صورت رابط کاربر و سخت افزار کامپیوتر عمل می کند. برای سیستم عامل انجام سه وظیفه را می توان در نظر گرفت:

۱- سهولت

سیستم عامل استفاده از کامپیوتر را برای کاربر ساده تر می کند.

۲- کارآمدی

سیستم عامل موجب استفاده کارآمد و بهینه از منابع سیستم کامپیوتری می شود.

۳- قابلیت رشد

سیستم عامل باید به نحوی ساخته شده باشد که توسعه آن میسر باشد.

سیستم عامل چیزی جز یک برنامه کامپیوتری نیست. این برنامه پردازنده را برای استفاده از سایر منابع سیستم هدایت می کند.

جایگاه سیستم عامل

می توان به سخت افزار و نرم افزاری که کاربردها را برای کاربر ارائه می کند به صورت لایه ای (شکل زیر) نگاه کرد. در این نگاه، سیستم عامل یک میانجی برای تسهیل دسترسی برنامه ساز و برنامه های کاربردی از امکانات و خدمات می باشد

برنامه های کاربردی
برنامه های سیستمی
سیستم عامل
سخت افزار

سخت افزار، پایین ترین سطح است که شامل سه قسمت است:

۱- دستگاه های فیزیکی (مانند سیم ها، منبع تغذیه، تراشه های مدار مجتمع)

۲- ریز معماری (شامل ثبات های درون CPU و یک مسیر داده شامل واحد محاسبه و منطق)

۳- زبان ماشین

تذکر: برنامه های سیستمی که در بالای سیستم عامل قرار دارند (مانند کامپایلرها، ادیتورها و ..)، بخشی از سیستم عامل نمی باشند، اگر چه توسط بعضی از سازندگان سیستم عامل ها عرضه می شوند.

سیستم عامل در زمینه های زیر خدمات خود را ارائه می دهد:

- ۱- ایجاد برنامه
- ۲- اجرای برنامه
- ۳- دسترسی به دستگاههای ورودی/خروجی
- ۴- کنترل دسترسی به پرونده ها
- ۵- دسترسی به سیستم عامل
- ۶- کشف و پاسخ به خطاها
- ۷- حسابداری.

سیستم عامل از طریق هسته (Kernel) ، با منابع سخت افزاری و نرم افزاری و از طریق پوسته (shell) با کاربران، ارتباط برقرار می کند.

بخشی از سیستم عامل که در حافظه اصلی قرار دارد را هسته سیستم عامل می گویند.

یک سیستم عامل ممکن است به دلایل زیر در طول زمان تغییر کند:

- ۱- ارتقاء و انواع جدید سخت افزار
- ۲- خدمات جدید
- ۳- رفع خطا

مولفه های سیستم عامل

سیستم بزرگی چون سیستم عامل را باید به مولفه های (components) کوچکتری تقسیم کرد. اکثر سیستم عامل ها دارای مولفه های زیر می باشند:

۱- مدیریت فرایند

برخورد با بن بست، ایجاد و حذف فرایندها، تعویق و از سرگیری فرایندها، هماهنگی فرایندها.

۲- مدیریت حافظه اصلی

تعیین بخشهای پر حافظه، تعیین فرایندی که باید لود شود، تخصیص حافظه و آزاد سازی حافظه.

۳- مدیریت حافظه ثانویه

مدیریت فضای آزاد، تخصیص حافظه، زمانبندی دیسک.

۴- مدیریت فایل

ایجاد و حذف فایلها و دایرکتوری ها، نگاشت فایلها در حافظه ثانویه و تهیه پشتیبان.

۵- مدیریت سیستم I/O

مدیریت بافرها، تخصیص کانالهای I/O و دستگاهها به فرایندها

تاریخچه سیستم‌های عامل

نسل اول: سیستم‌های ردیفی

در اولین کامپیوترها (حدوداً ۶۰ سال قبل)، برنامه ساز مستقیماً با سخت افزار در تراکنش بود و سیستم عاملی در کار نبود. این ماشینها از طریق چراغهای نمایش، کلیدها و نوعی دستگاه ورودی و چاپگر اجرا می شدند. مساله اصلی این سیستمها، زمانبندی و زمان نصب بود. چون در این عملیات، کاربران به صورت ردیفی و یکی پس از دیگری به کامپیوتر دسترسی داشتند، آن را پردازش ردیفی می خوانیم.

نسل دوم: سیستم‌های دسته‌ای ساده

زمانی که به خاطر زمانبندی و نصب در سیستمهای ردیفی به هدر می رفت قابل قبول نبود. برای افزایش استفاده از ماشین، مفهوم سیستم عامل دسته ای به وجود آمد. اولین سیستم عامل دسته ای در اوسط دهه ۱۹۵۰ به وجود آمد. ایده اصلی، استفاده از نرم افزاری به نام ناظر بود. با استفاده از سیستم عامل دسته ای، دیگر کاربر دسترسی مستقیم به ماشین ندارد و کار خود را روی کارت یا نوار به متصدی کامپیوتر می دهد. اپراتور کارها را به صورت ردیفی دسته کرده و همگی را روی یک دستگاه ورودی می گذارد تا مورد استفاده ناظر قرار گیرد. مساله زمانبندی و زمان تنظیم شرایط اولیه کارها به عهده ناظر است.

با هر کار، دستورالعملهایی از زبان کنترل کار (JCL) نیز وجود دارد. زبان کنترل کار، نوعی زبان برنامه نویسی برای فرمان دادن به ناظر می باشد.

JCL : Job Control Language

اسپولینگ

در سیستم‌های اولیه، CPU گرانترین جزء کامپیوتر بود و افزایش راندمان CPU مهمترین محرک در تکامل سیستم‌های عامل بوده است. برای همین منظور، تکنیک Spooling مطرح شد. اسپولینگ I/O یک کار را با محاسبات کار دیگر به طور همزمان انجام می دهد. حتی می تواند در حین خواندن ورودی‌های یک کار، خروجی‌های کار دیگر را انجام دهد. اسپولینگ موجب می شود تا CPU و دستگاههای I/O با سرعت بیشتری کار کنند. استفاده از spooling در سیستم عامل دسته ای باعث افزایش سرعت می شود.

سیستم‌های دسته ای Offline Spooling

در این سیستم‌ها، کارها توسط دستگاه کارت خوان یک کامپیوتر آهسته و ارزان خوانده می شد و به کمک یک نوار گردان بر روی یک نوار مغناطیسی ذخیره می شدند. بعد از تشکیل یک دسته از کارها، نوار توسط اپراتور به CPU اصلی منتقل شده تا اجرا شروع شود. نتیجه اجرا بر روی نوار دیگری نوشته می شد. بعد از

اجرای تمام دسته کارها، اپراتور نوار خروجی را به یک پردازنده آهسته منتقل می‌کرد تا عمل چاپ انجام شود. در این روش ارتباط پردازنده اصلی محاسباتی با کارت خوان و چاپگر، غیر مستقیم بود.

فواید سیستم‌های دسته ای offline Spooling نسبت به سیستم‌های قبلی

- ۱- افزایش راندمان پردازنده گران قیمت
- ۲- عملیات ساده‌تر
- ۳- ساده شدن استفاده از سیستم از راه دور.

معایب سیستم‌های دسته ای Offline Spooling

- ۱- تاخیر بین زمان تحویل کار و زمان تکمیل کار
- ۲- نیاز به سخت افزار اضافی
- ۳- عدم وجود اولویت
- ۴- استفاده زیاد از دستگاههای جانبی

سیستم‌های دسته ای Online Spooling

با ظهور دیسک‌های سخت، سیستم‌های دسته ای متحول شدند. در این سیستم‌ها بلافاصله بعد از ورود کارها، کارت‌ها خوانده شده و به دیسک منتقل می‌شدند. زمانی که اجرای یک کار تمام می‌شود، سیستم عامل با سرعت بیشتری یک کار جدید را از روی دیسک برداشته و به حافظه اصلی لود کرده و سپس آن را اجرا می‌کند.

در خروجی نیز از اسپولینگ استفاده می‌شود. اسناد خروجی ابتدا بر روی دیسک ذخیره شده و زمانی که خروجی قبلی چاپگر تمام شد، این اسناد از دیسک به بافر چاپگر منتقل می‌شوند. در این تکنیک که Spooling (عملیات پیوسته، مستقیم و همزمان دستگاه‌های جانبی) نامیده می‌شود، ارتباط CPU با دستگاه‌های جانبی مستقیم است و نیازی به پردازنده‌های آهسته و نوار گردان‌های اضافی و حمل نوارها توسط اپراتورها نمی‌باشد.

SPOOL : Simultaneous Peripheral Operation On-Line

مزایای اسپولینگ online نسبت به offline عبارتند از :

- ۱- دسترسی با اولویت
- ۲- گردش سریع کارها
- ۳- بالا بودن راندمان CPU و دستگاههای I/O
- ۴- امکان داشتن همزمان چند مدرک ورودی/خروجی

از اسپولینگ برای پردازش داده‌های راه دور می‌توان استفاده کرد. CPU داده‌ها را از طریق مسیرهای ارتباطی به چاپگر راه دور می‌فرستد یا ورودی‌ها را از یک کارت خوان راه دور دریافت می‌کند. در پردازش راه دور CPU دخالت ندارد و فقط مواظب است که چه هنگام کار تمام شود تا دسته دیگری از کارها را جمع‌آوری کند.

نرم افزار spooler کارها را دسته بندی می کند. این نرم افزار اجزاء یک کار (داده، برنامه و JCL) را از هم جدا می کند. سپس JCL ها دستور به دستور اجرا شده و توسط سیستم اسپولینگ روی دیسک اصلی قرار می گیرند و جدولی به نام ISPT (Input Spool Table) ساخته می شود. هر سطر این جدول در برگیرنده اطلاعات کنترلی یک Job می باشد.

نسل سوم: سیستم های دسته ای چند برنامه گی

حتی با ردیف کردن خودکار کارها به وسیله سیستم عامل ساده دسته ای، به علت کند بودن دستگاههای I/O در مقایسه با پردازنده، پردازنده اکثرا بی کار است. تمام کارهایی که وارد سیستم می‌شوند، در انبار کار (job pool) نگهداری می‌شوند. این کارها شامل فرایندهایی می‌باشند که بر روی دیسک قرار دارند و منتظر تخصیص حافظه می‌باشند. اگر چند کار آماده ورود به حافظه وجود داشته باشند، سیستم یکی را انتخاب کرده (زمانبندی کار) و آنرا به حافظه می‌آورد تا اجرا کند. همچنین اگر چند کار به طور همزمان آماده اجرا باشند، سیستم یکی از آنها را انتخاب می‌کند و CPU را به آن تخصیص می‌دهد. (زمانبندی CPU). ایده چند برنامه‌ای به این صورت است که سیستم عامل در هر زمان چند کار را در حافظه نگه می‌دارد و یکی از کارهای موجود در حافظه را انتخاب کرده و اجرای آن را شروع می‌کند. اگر این کار منتظر عمل دیگری مانند آماده شدن نوار یا تکمیل یک عمل I/O باشد، CPU به اجرای کار دیگری مشغول می‌شود و اگر آن کار نیز نیاز به انتظار داشته باشد، به کار دیگری می‌پردازد و این روند ادامه می‌یابد. با پایان مدت انتظار کار اول، دوباره CPU در اختیار آن قرار می‌گیرد. این عملیات باعث می‌شود که CPU بیکار نماند.

سیستم های دسته ای برای اجرای برنامه های بزرگ که نیاز به محاوره کمی دارند، مناسب می باشند.

مشکلات سیستم دسته ای از نظر کاربر

- ۱- کاربر برای به دست آوردن نتایج مجبور است از کارتهای کنترلی استفاده کند.
- ۲- مراجعه بعدی در کارهای چند مرحله ای ممکن است به نتایج مراحل قبلی وابسته باشد.
- ۳- برنامه نویس نمی تواند برنامه در حال اجرا را تغییر دهد. (دیباگ برنامه ها ایستا می باشد)

ممکن است حافظه را برای نگهداری ۳،۴ یا تعداد بیشتری از برنامه‌ها گسترش دهیم و پردازنده به تمام آنها پردازد. این عمل را چند برنامه‌ای یا چند وظیفه‌ای می‌گویند که موضوع اصلی سیستم‌های عامل امروزی است.

نسل چهارم : سیستم‌های اشتراک زمانی (Time sharing)

سیستم‌های اشتراک زمانی (چندبرنامگی تعاملی) توسعه منطقی سیستم‌های چند برنامه‌ای می‌باشند. در سیستم عامل اشتراک زمانی، وقت پردازنده بین کارها به اشتراک گذاشته می‌شود. برای اجرای چند کار، پردازنده بین آنها سوئیچ می‌کند و کاربران می‌توانند با برنامه‌های در حال اجرا تعامل داشته باشند. در سیستم‌های اشتراک زمانی ارتباط کاربر با سیستم به صورت محاوره‌ای (interactive یا online) است. سیستم‌های اشتراک زمانی، هزینه استفاده محاوره‌ای از سیستم‌های کامپیوتری را معقول کرده‌اند و با استفاده از زمانبندی CPU و چند برنامه‌ای، به هر کاربر زمان محدودی را تخصیص داده و هر کاربر حداکثر یک برنامه در حافظه دارد.

در سیستم‌های اشتراک زمانی، چون سیستم به سرعت از برنامه‌ای به برنامه دیگر می‌رود، هر کاربر فکر می‌کند که یک کامپیوتر اختصاصی در اختیار اوست، در حالی که یک کامپیوتر بین همه کاربران مشترک است.

خصوصیات سیستم‌های اشتراک زمانی عبارتند از:

- ۱- پیچیده‌تر از چند برنامه‌ای هستند.
- ۲- برای مواردی که نیاز به زمان پاسخ کوتاه است، استفاده می‌شوند.
- ۳- امکان چند برنامگی را فراهم می‌سازد.
- ۴- وجود یک سیستم فایل ضروری است.
- ۵- در main frame قابلیت چند کاربره و چند برنامگی را فراهم می‌کند.

چندبرنامه‌ای دسته‌ای و اشتراک زمانی هر دو از چندبرنامگی استفاده می‌کنند.

تفاوت‌های اصلی چندبرنامه‌ای دسته‌ای با اشتراک زمانی

اشتراک زمانی	چندبرنامه‌ای دسته‌ای	
حداقل زمان پاسخ	حداکثر استفاده از پردازنده	هدف اصلی
فرمانهایی که از پایانه وارد می‌شود.	دستورالعمل‌های JCL که همراه کار ارائه شده است.	منبع دستورات به سیستم عامل

تذکر: سیستم‌های زیر در کتاب سیلبرشاتس آورده شده است:

۱- سیستم‌های موازی (یا چند پردازنده ای)

۲- سیستم‌های عامل توزیع شده (گسترده)

۳- سیستم‌های عامل بی‌درنگ (Real time)

سیستم‌های موازی (یا چند پردازنده ای)

سیستم‌هایی که بیش از یک پردازنده در آنها وجود دارد را چند پردازنده ای (Multi Processor) می‌نامند. پردازنده‌ها در این سیستم با یکدیگر ارتباط نزدیکی دارند و از گذرگاه آدرس، ساعت و گاهی حافظه و دستگاه‌های جانبی به طور اشتراکی استفاده می‌کنند. این سیستم‌ها را اتصال محکم (tightly coupled) نیز می‌نامند.

انواع سیستم‌های چند پردازنده‌ای

الف - چندپردازنده‌ای متقارن

در سیستم چندپردازنده‌ای متقارن (Symmetric)، هر پردازنده از کپی یکسانی از سیستم عامل استفاده می‌کند که این کپی‌ها در صورت لزوم با یکدیگر ارتباط برقرار می‌کنند.

ب - چندپردازنده‌ای نامتقارن

در سیستم چند پردازنده‌ای نامتقارن، هر پردازنده کار خاصی را انجام می‌دهد. کنترل سیستم به عهده پردازنده اصلی می‌باشد و پردازنده‌های دیگر منتظر دستور پردازنده اصلی هستند یا کار از قبل تعیین شده‌ای دارند. این طرح رئیس / مرئوس (master/slave) را بیان می‌کند، که پردازنده اصلی (master)، کارها را برای پردازنده‌های دیگر (slave)، زمانبندی کرده و به آنها تخصیص می‌دهد.

مزایای سیستم متقارن نسبت به نامتقارن عبارتند از :

۱- قابل حمل بودن روی سیستم‌های سخت افزاری مختلف

۲- تعادل بار سیستم به علت اجرای سیستم عامل روی چند پردازنده

دلایل ساخت سیستم‌های چند پردازنده‌ای

۱- افزایش توان عملیاتی (throughput) : تعداد کارهای انجام شده در یک واحد زمانی.

۲- افزایش قابلیت اعتماد

(خرابی یک پردازنده منجر به از کار افتادن سیستم نمی‌شود بلکه از سرعت آن می‌کاهد).

۳- مقرون به صرفه بودن از نظر اقتصادی

سیستم‌های عامل توزیع شده (گسترده)

در این سیستم‌ها، محاسبات بین چند پردازنده توزیع می‌شود. هر پردازنده، حافظه و ساعت مخصوص به خود را دارد و از طریق خطوط ارتباطی با یکدیگر مرتبط‌اند. همچنین پردازنده‌ها از نظر اندازه و عملکرد با

یکدیگر فرق دارند. سیستم‌های توزیعی را سیستم‌های ارتباط ضعیف (Loselycoupled) نیز می‌گویند.

دلایل ساخت سیستم‌های توزیعی

الف- اشتراک منابع : کاربری در یک سایت می‌تواند از چاپگری در سایت دیگر استفاده کند.

ب- قابلیت اعتماد: خرابی یک کامپیوتر، دیگری را تحت تاثیر قرار نمی‌دهد.

ج- افزایش سرعت محاسبات: توزیع یک محاسبه در بین چند سایت

د- ارتباطات

سیستم‌های عامل بی‌درنگ (Real time)

سیستم عامل بی‌درنگ (بلادرنگ)، نوعی از سیستم‌های عامل‌های همه منظوره می‌باشد و در صورتی به کار گرفته می‌شود که برای عملکرد یک پردازنده نیاز به زمان دقیقی باشد. یک سیستم بی‌درنگ وقتی درست کار می‌کند که در محدودیت زمانی مشخص، نتایج مورد انتظار را تولید کند. یعنی پردازش باید در محدودیت زمانی خاص انجام شود وگرنه سیستم از کار می‌افتد.

سیستم‌های نظامی، تزریق سوخت اتومبیل، کنترل کننده‌های لوازم خانگی، کنترل صنعتی و تصویرسازی پزشکی نمونه‌هایی از سیستم‌های بی‌درنگ می‌باشند. سیستم‌های بی‌درنگ بر دو نوع نرم و سخت می‌باشند.

ویژگی‌های سیستم‌های بی‌درنگ نرم (Soft Real Time)

- ۱- دارای محدودیت زمانی نسبتاً دقیقی می‌باشند.
- ۲- در پروژه‌های علمی پیشرفته استفاده می‌شوند.
- ۳- با سیستم‌های دیگر قابل ترکیب می‌باشند.
- ۴- مهلت زمانی را پشتیبانی نمی‌کنند. (از آنها در کنترل صنعتی و روبات‌ها استفاده نمی‌شود).

ویژگی‌های سیستم‌های بی‌درنگ سخت (Hard Real Time)

- ۱- دارای محدودیت زمانی دقیقی می‌باشند و کارهای بحرانی به موقع انجام می‌شوند.
- ۲- تمام تاخیرهای موجود در سیستم باید از بین بروند.
- ۳- از ویژگیهای پیشرفته سیستم عامل استفاده نمی‌شود.
- ۴- با سیستم‌های اشتراک زمانی برخورد (Conflict) دارند و با یکدیگر ترکیب نمی‌شوند.
- ۵- هیچکدام از سیستم‌های همه منظوره موجود از عملکرد بی‌درنگ سخت پشتیبانی نمی‌کنند.

انواع سیستم عامل از نظر ساختاری

انواع سیستم عامل از نظر ساختاری عبارتند از:

- ۱- یکپارچه (Monolithic)
- ۲- لایه ای (Layered)
- ۳- ماشین مجازی (Virtual Machine)
- ۴- مشتری - خدمتگذار (Client /Server)

ساختار یکپارچه

ساده ترین ساختار برای سیستم عامل است که در DOS از آن استفاده می شود. در این ساختار واسط ها و سطوح عملکرد به خوبی از هم تفکیک نشده اند و برنامه های کاربردی می توانند به روالهای I/O دستیابی داشته باشند و مستقیماً بر روی مونیتر یا دیسک بنویسند.

ساختار لایه‌ای

با اعمال خاصیت پیمانه ای بودن به سیستم عامل، سیستم عامل می تواند کنترل بیشتری بر روی کامپیوتر و برنامه های کاربردی داشته باشد. یکی از بهترین روشها برای این کار، روش لایه ای است که سیستم عامل را به تعدادی لایه تقسیم می کند و هر کدام بر روی یکدیگر قرار می گیرند. لایه پایینی (شماره صفر)، سخت افزار است و لایه بالایی واسط کاربر است. طراحی و پیاده سازی ساختار لایه ای ساده است.

ویژگی های ساختار لایه ای

- ۱- هر لایه فقط از توابع و خدمات لایه های پایین تر استفاده می کند که باعث ساده شدن خطایابی می شود.
- ۲- خطایابی از لایه پایین شروع می شود و سپس لایه بعدی خطایابی می شود. اگر در حین خطایابی لایه ای، خطایی یافت شود، می دانیم که خطا در همان لایه است، زیرا لایه های پایینی قبلاً خطایابی شده اند.
- ۳- هر لایه می تواند عملیات، ساختمان داده ها و سخت افزار را از لایه های بالاتر مخفی کند.

مشکلات روش لایه ای عبارتند از:

- ۱- هر لایه نیاز به برنامه ریزی دقیقی دارد، چون فقط می تواند از لایه های زیرین استفاده کند.
- ۲- پایین بودن کارایی نسبت به روشهای دیگر. (تولید سربار زیاد)

مزایای ساختار لایه ای نسبت به یکپارچه عبارتند از:

- ۱- قابلیت گسترش بیشتر
- ۲- خطایابی ساده تر
- ۳- مدیریت ساده تر

ساختار ماشین مجازی (Virtual Machine)

یک سیستم کامپیوتری دارای لایه های سخت افزار، هسته و برنامه های سیستم می باشد. برنامه های سیستم که در بالای هسته قرار دارند، می توانند از فراخوانیهای سیستم و دستورات سخت افزاری استفاده کنند. بعضی سیستمها از این الگو تبعیت می کنند و حتی برنامه های سیستم می توانند از طریق برنامه های کاربردی فراخوانی شوند و برنامه های سیستم می توانند آنچه را که در زیر آنها قرار دارد را مشاهده نمایند، به طوری که گویی برنامه های کاربردی بخشی از خود ماشین هستند. این شیوه لایه ای را ماشین مجازی می نامند که یک کپی از کامپیوتر در اختیار هر فرایند قرار می دهد. اجرا شدن سیستم عامل DOS تحت محیط ویندوز با استفاده از ایده ماشین مجازی انجام گرفته است.

مزایای ساختار ماشین مجازی

- ۱- امکان اجرای هم زمان چند سیستم عامل مختلف.
- ۲- شبیه سازی دستورات ماشینی که کامپیوتر فاقد آن است.
- ۳- امکان برنامه نویسی به زبان های غیر اسمبلی.
- ۴- تست سیستم عامل جدید.

با تغییراتی در روش ماشین مجازی، ساختاری به نام **Exokernels** به وجود آمده است. در این ساختار نیازی به لایه نگاشت نمی باشد و همچنین برنامه ای در پایین ترین لایه در مد هسته اجرا شده که منابع را به ماشین های مجازی تخصیص داده و بررسی می کند که ماشینی از منابع ماشین دیگر استفاده نکند.

ساختار مشتری – خدمتگزار (Client – Server)

در این ساختار، اکثر وظایف سیستم عامل در سطح کاربر انجام می شود و هسته از طریق پیام ها بین مشتری/خدمتگزار ارتباط برقرار می سازد. ایده طراحی این ساختار، کمینه کردن هسته و انتقال کدها به لایه های بالاتر می باشد. مزایای ساختارهای مشتری – خدمتگزار عبارتند از:

- ۱- طراحی ساده تر سیستم عامل
- ۲- استفاده در سیستم های توزیعی
- ۳- عدم خرابی کل سیستم در صورت خرابی یک سرور