



سیستم عامل پیشرفته

درس سیستم عامل پیشرفته به عنوان یکی از دروس اصلی دو گرایش نرم افزار و سخت افزار در رشته کارشناسی ارشد کامپیوتر مورد توجه قرار می گیرد. در جلسه اول از این درس، ابتدا منابع درسی و سپس یک چهار چوب کلی از مطالبی که در این درس مطرح می شود و در نهایت مقدمات درس را بررسی خواهیم کرد. ۳ کتاب عمده به عنوان مراجع و در واقع منابع اصلی این درس معرفی خواهد شد.

کتاب اول Modern Operating system نوشته A.S.Tanenbaum می باشد. دو edition از این کتاب موجود است، که در واقع مباحث مطرح شده بر مبنای edition دوم از انتشارات Prentice Hall خواهد بود.

کتاب دوم که در واقع در سال ۲۰۰۲ توسط انتشارات Prentice Hall مطرح شده است با نام Distributed Systems نوشته A.S.Tanenbaum می باشد. مطالب عنوان شده از این دو کتاب، سازماندهی و ارائه می شود.

کتاب سوم که جهت مطالعه بیشتر می باشد، کتابی با نام Distributed Operating system نوشته A.S.Tanenbaum می باشد.

در مرحله اول کلیات و چهار چوب مباحثی که در درس سیستم عامل پیشرفته ارائه خواهد شد، را مرور می کنیم، در ابتدا ۶ فصل اول کتاب Modern Operating system به صورت مختصر و سریع به عنوان یک مرور کلی از درس سیستم عامل ۱، بیان می گردد. در ادامه سیستم های عامل چند رسانه ای یا Multimedia Operating systems مورد بررسی قرار خواهد گرفت، مرجع اصلی این مبحث، فصل ۷ کتاب Modern Operating system می باشد. مبحث دیگر مفاهیم سیستم های عامل توزیع شده یا Distributed systems خواهد بود، که فصل هشتم کتاب Modern Operating system و ۳ فصل اول کتاب Distributed Systems مباحث مطرح شده را در بر می گیرد. مبحث آخر، در بردارنده مباحث تکمیلی سیستم های توزیع شده می باشد.

نکته: منبع سوم به نوعی تمامی مطالب را به صورت جامع در خود جای داده است و می توان از آن برای مطالعه بیشتر در غالب منبعی که دارای مباحث مفیدی در بخش سوالات و تمرینات و همچنین برای امتحانات خواهد بود، استفاده کرد.

فصل اول: مقدمه (کتاب Modern Operating system)

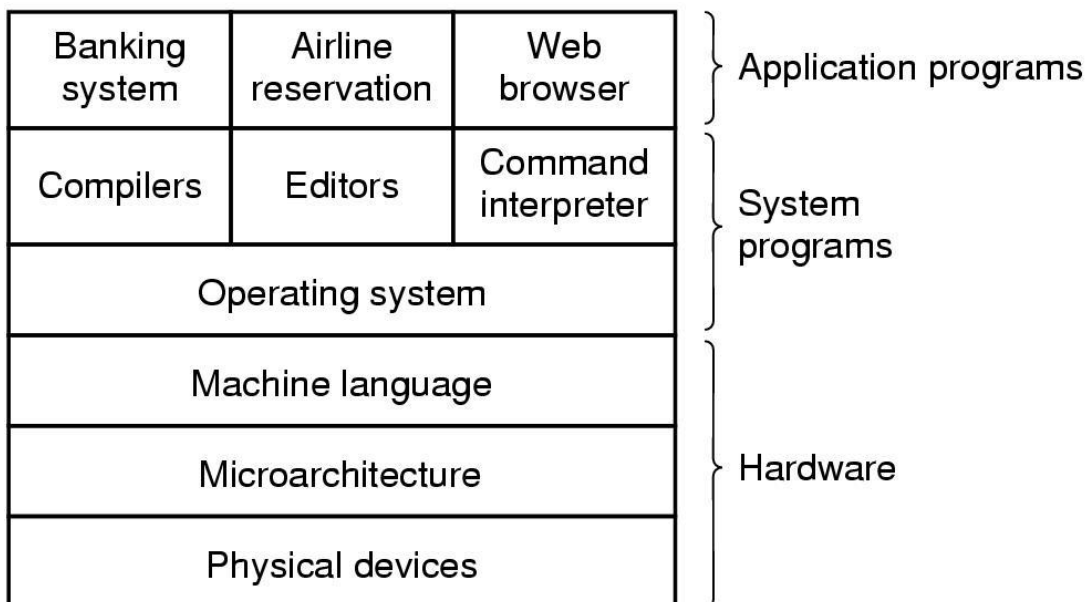
در این فصل مروری بر مفاهیم سخت افزار کامپیوتر یا computer hardware و همچنین مفاهیم مهم در سیستم های عامل صورت می گیرد. سپس مطالبی از فراخوانی های سیستمی یا system call ها در سیستم های عامل و در انتها ساختارهای معروف و مطرح در سیستم های عامل و مزایا و معایب آنها عنوان خواهد شد.

اصولاً یک سیستم کامپیوتری از چه اجزایی تشکیل شده است؟ یک سیستم کامپیوتری از سه جزء اصلی تشکیل شده است:

۱- سخت افزار (hardware)

۲- برنامه های سیستمی (System programs)

۳- برنامه های کاربردی (Application Programs)



همانطور که در شکل مشاهده می شود ارتباط این سه جزء با هم یک ارتباط لایه ای است یعنی **system program** به عنوان یک لایه بر روی **hardware** قرار می گیرد و **Application program** ها معمولاً به عنوان یک لایه بر روی **system program** ها قرار می گیرند.

Hardware

سخت افزار اجزایی دارد که برای استفاده از این لایه به وجود این اجزاء نیازمندیم.

اجزای لایه سخت افزار به سه گروه تقسیم می شود:

۱- دستگاه های فیزیکی (physical Devices)

۲- معماری ریز (Microarchitecture)

۳- زبان ماشین

با ترکیب این سه جزء شما قادر به استفاده از سخت افزار خواهید بود.

همانطور که می دانید تمامی سخت افزارها هر سه این اجزاء را ندارند به عنوان مثال ممکن است که بعضی از سخت افزارها **Wired** باشند و از مفاهیم **Microarchitecture** استفاده نکرده باشند.

Application programs

برنامه های کاربردی مانند **Web browser** ها، نمایشگرهای اینترنتی، **Airline reservation** (سیستم هایی که برای رزرو بلیط استفاده می شوند یا احیاناً سیستم های بانکی)

System programs

یک جزء عمده یا یک نوع عمده از برنامه های سیستمی را غالباً به سیستم های عامل تخصیص می دهیم و به عبارت دیگر ما سیستم عامل را یک **System program** می دانیم که در ارتباط تنگاتنگی با برنامه های سیستمی است. مانند کامپایلرها، ادیتورها، مفسرها و مواردی از این قبیل.

مطالب عنوان شده در بالا به این مطلب اشاره دارد که سیستم عامل در گروه **System program** ها یا برنامه های سیستم قرار دارد. این مطلب مشخص خواهد کرد که سیستم عامل یک نرم افزار است و برنامه ای است که نوشته شده است که نویسنده این برنامه باید با توجه به جایگاه آن در این مدل اطلاعات جامع از سخت افزار داشته باشد تا بتواند برنامه ای را بنویسد که در واقع بر روی سخت افزاری قرار می گیرد و بنابراین برای برقراری ارتباط سیستم عامل با سخت افزار، نویسنده یک سیستم عامل باید جزئیات مربوط به زبان ماشین را خوب بداند.

سیستم عامل چیست ؟

تعریف های گوناگونی برای یک سیستم عامل مطرح می شود و در واقع از یک دیدگاه ساده، سیستم عامل، یک برنامه بسیار بزرگ است که وظایف خاصی را بر عهده دارد. از جمله این وظایف مدیریت منابع می باشد. مدت زمان (طول عمر) این برنامه از لحظه روشن شدن یک ماشین و یا به عبارت دقیق تر لحظه راه اندازی یک ماشین تا لحظه خاموش شدن آن است و در کل در این بازه سیستم عامل در حال فعالیت است و منابع را کنترل می کند.

از جمله دلایل استفاده از سیستم عامل و به عبارتی ضرورت استفاده از آن، استفاده اقتصادی از منابع است. منظور از استفاده اقتصادی از منابع، افزایش بهره وری منابع و حتی الامکان کاهش زمان بیکاری منابع است و در عین حال سیستم عامل باعث خواهد شد که یک سطح دسترسی بالاتر برای کاربران فراهم شود و کاربران نیاز به دانستن جزئیات سخت افزاری نداشته باشند.

تعاریف دیگر در ارتباط با سیستم عامل :

سیستم عامل یک ماشین توسعه یافته است. این توسعه یافتگی که در طراحی ماشین اتفاق افتاده است باعث می شود بسیاری از جزئیاتی که در داخل ماشین باید انجام شود از دید کاربر پنهان باشد. سیستم عامل با ارائه یک ماشین مجازی باعث شده است کاربر، ماشین را به شکل یک ماشین مجازی ببیند و به این ترتیب استفاده از ماشین آسان تر خواهد شد.

از دیدگاه دیگر سیستم عامل یک **Resource manager** یا یک مدیر منابع است. به این معنی که هر برنامه برای این که بتواند از منبعی استفاده کند باید زمان مورد استفاده از منبع را به سیستم عامل اعلام کند

و در عین حال فضای مورد نظر بر روی آن منبع یا خود منبع، باید از طریق سیستم عامل به برنامه درخواست کننده آن منبع تخصیص داده شود.

تاریخچه سیستم های عامل

به عنوان اولین مثال از یک سیستم عامل یا به عبارتی یک سیستمی که حداقل با مفاهیم مشابه سیستم عامل کار می کند می توان سیستم دسته ای یا Batch system را بیان کرد.

تعریف کلی از سیستم های دسته ای

سیستم ها به دو گروه تقسیم می شوند:

۱- سیستم های دسته ای (batch system)

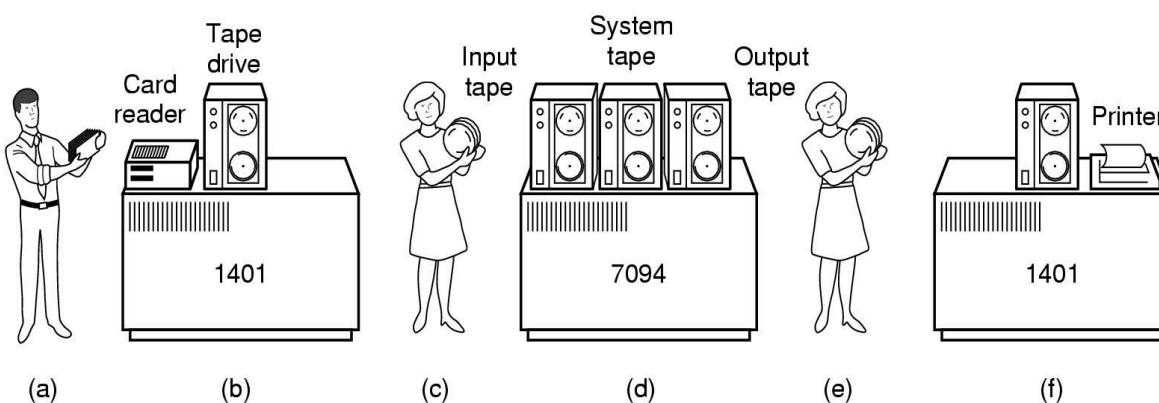
۲- سیستم های محاوره ای (Interactive system)

سیستم های دسته ای به سیستم هایی گفته می شود که پردازش داده ها در آنها به صورت دسته ای انجام می شود. برای اینکه تعریف دسته ای بهتر مشخص شود سیستم های محاوره ای را نیز می توان تعریف کرد.

سیستم های محاوره ای سیستم هایی هستند که برای پیشرفت در انجام پردازش خود نیاز به ارتباط با کاربر دارند یعنی باید در طی پیشرفت و در طی جلو رفتن این فرآیند پردازشی، با کاربر تعامل داشته باشد.

سیستم های دسته ای سیستم هایی هستند که معمولاً ورودی ها را به یکباره می گیرند پردازش ها را انجام می دهند و بعد از اینکه پردازش کامل شد خروجی را به یکباره تولید کرده و در اختیار کاربر قرار می دهد. حال به مفهوم سیستم های Offline spooling به عنوان یکی از سیستم های به نوعی دسته ای و سیستم هایی که سعی شده است مفاهیم سیستم عاملی در آن رعایت شود می پردازیم.

سیستم‌های Offline spooling



همانطور که در شکل دیده می شود از سه جزء اصلی تشکیل شده است، ۲ جزء ابتدا و انتهای آن، کامپیوترهای قدیمی ۱۴۰۱ هستند و یک جزء وسطی که کامپیوتر مدل ۷۰۹۴ است. روال کلی این سیستم ها به این گونه بود که دستگاه کارتخوان وظیفه وارد کردن اطلاعات را در سیستم داشت. بنابراین در گام اول ورودی‌ها از طریق دستگاه کارتخوان وارد سیستم می شدند و سپس بر روی یک Tape drive ذخیره می شدند آنگاه Tape drive ای که شامل اطلاعات ورودی‌ها بود از ماشین ۱۴۰۱ به ماشین دوم منتقل می شد.

چرا ماشین ۱۴۰۱ / استفاده می شد؟

علت عمده قضیه این است که ۱۴۰۱ ها ماشین هایی ارزان قیمت بودند و پردازنده گرانی نداشتند از سوی دیگر معمولاً دستگاه‌های کارتخوان دستگاه‌هایی بودند که زمان اجرا یا زمان فعالیت آنها طولانی بود. بنابراین برای اینکه یک CPU گران قیمت که در ماشین ۷۰۹۴ قرار داشت اتلاف نشود از ماشین‌های ۱۴۰۱ با پردازنده‌های ارزان قیمت تر استفاده شد و این ماشین ها در واقع اطلاعات موجود بر روی کارت ها را روی Tape drive ها منتقل می کردند که در آن زمان Tape drive ها یکی از رسانه‌های نسبتاً سریع بود. امروزه دیگر Tape drive مورد استفاده قرار نمی گیرد. ولی در عین حال، هدف مدلی است که در روند اجرای این سیستم بکار گرفته شده است. بنابراین دستگاه ورودی جدا بود، دستگاه کارتخوان ورودی ها را می خواند روی یک نوار قرار می داد، یک انسان این نوار ها را به ماشین ۷۰۹۴ که در واقع وظیفه

پردازش را بر عهده داشت منتقل می‌کرد. لذا فرض بر این قرار می‌گرفت که با وجود نوارها دستگاه پردازشگر که از یک پردازنده قوی برخوردار است مدت زمان زیادی را منتظر خواندن ورودی‌ها نخواهد شد و زمان بیکاری CPU بسیار زیاد نخواهد بود، بعد از اینکه پردازش انجام می‌شد مجدداً خروجی‌ها بر روی یک نوار نوشته می‌شد و سپس از این نوارها جهت انتقال داده‌ها به ماشین ۱۴۰۱ دیگری که وظیفه انتقال خروجی را بر عهده داشت استفاده می‌شد.

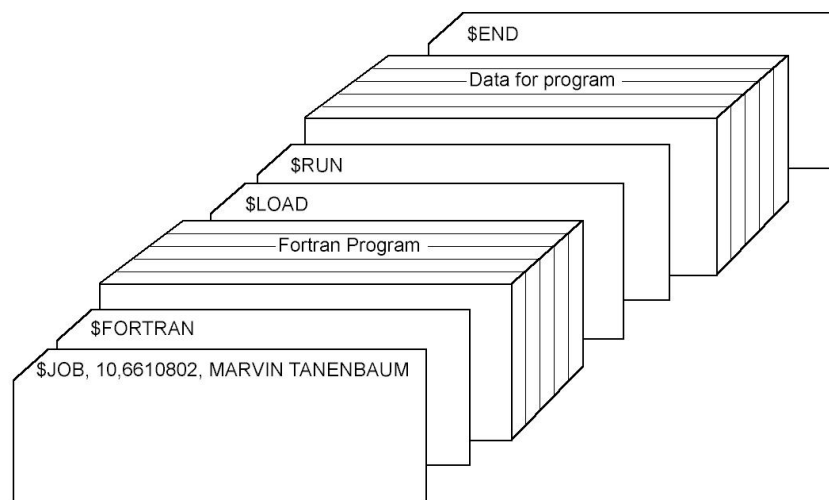
هدف از این مثال مطرح شده آن است که برای استفاده بهینه از CPU در سیستمی که وظیفه پردازش را بر عهده دارد باید یک سیستم بهینه برای آن طراحی کرد، به عبارت دیگر نباید CPU ها مدت زمان زیادی را منتظر انجام برخی عملیات که معمولاً زمان انجام این عملیات طولانی است صرف کنند. لذا راه کاری که در سیستم **Offline spooling** بکار رفت این بود که برای خواندن ورودی‌ها از دستگاه‌های ارزان قیمت استفاده شود و در عین حال چه بسا اینکه ورودی‌های مختلف می‌توانستند توسط افراد مختلف از دستگاه‌های ورودی مختلف جمع‌آوری شوند و به تنها یک دستگاه پردازشگر منتقل شوند.

بعد از سیستم‌های **Offline spooling** سیستم‌های **Online Spooling** مطرح شد، در سیستم‌های **Online Spooling** این ۳ دستگاه ورودی تبدیل به یک دستگاه ورودی شد و ورودی‌ها که توسط دستگاه کارتخوان خوانده می‌شد **Spool** می‌شد و بر روی یک دستگاه ذخیره ساز دیگری به نام **hard disk** ذخیره می‌شد، لذا مدل شبیه مدل **Offline spooling** بود.

نتیجه‌ای که از این قسمت استنباط می‌شود این است که عملیات I/O که معمولاً زمان پردازش زیادی را شامل می‌شود نباید باعث ایجاد خلل در عملیات مربوط به پردازشگر و یا CPU شود. دیدگاه دیگر در ارتباط با تاریخچه سیستم‌های عامل به این مطلب می‌پردازد که از آنجاییکه سیستم عامل روی لایه مستقیم سخت افزار قرار گرفته است، باید مفاهیم و سلسله مراتب ایجاد سخت افزار را در پیشرفت سیستم‌های عامل دخیل نماییم. به عنوان مثال نسل‌های مختلف سخت افزاری و ایجاد کامپیوترهای مختلف هر کدام باعث ایجاد نسل‌هایی از سیستم‌های عامل شده‌اند، مثلاً اگر اولین نسل سیستم‌های کامپیوتری یا **First generation** را در نظر بگیریم (این نسل را از سال ۱۹۴۵ تا ۱۹۵۵ می‌دانستند) که مربوط به کامپیوترهای مبتنی بر **Board** های **Plug** و لامپ‌های خلاء بودند. نسل دوم (این نسل را از سال ۱۹۵۵ تا ۱۹۶۵ می‌دانستند) نسلی بودند که مبتنی بر ماشین‌های ترانزیستوری و سیستم‌های دسته‌ای بودند. سومین نسل سخت افزار یا ماشین‌های ایجاد شده (از سال ۱۹۶۵ تا ۱۹۸۰ مطرح بودند) نسلی از کامپیوترهای مبتنی بر مفاهیم **multi programming** و IC ها بودند. آخرین نسلی که هم اکنون در آن به سر می‌بریم که

آغاز آنها از سال ۱۹۸۰ تا روزهای کنونی است PC ها می‌باشند. در هر حال می‌دانید که هر کدام از ماشین ها، سیستم‌های عامل خاص خود را دارند که بعضاً این سیستم‌های عامل با هم سازگار نیستند. چه بسا در مواقعی برنامه‌هایی که برای برخی از این سیستم‌های عامل طراحی شده است را روی سیستم‌های عامل دیگری نتوانید منتقل کنید.

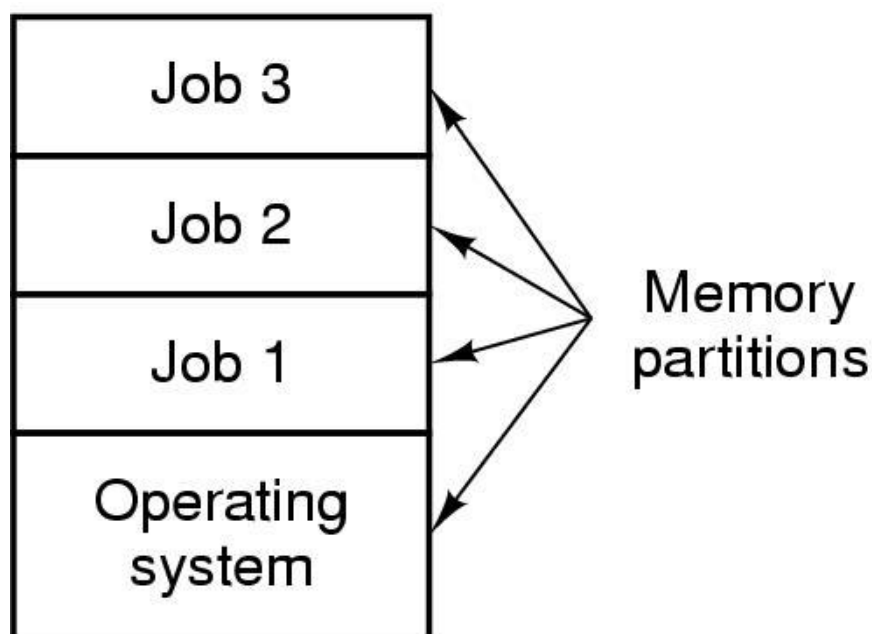
بنابراین از دیدگاه دیگری تاریخچه سیستم‌های عامل به نسل‌های کامپیوترها نیز بازخواهد گشت. به عنوان مثالی از تاریخچه سیستم‌های عامل می‌توان به سیستم‌های عامل دسته‌ای اشاره نمود. هدف آنها استفاده از سیستم‌های کارت خوان و بهینه استفاده کردن از این سیستم‌هاست. در مباحث مربوط به نسل‌ها عنوان شد که دومین نسل کامپیوترها این مفاهیم را پشتیبانی می‌کند.



در شکل مجموعه‌ای از کارت‌ها را می‌بینید که ضمن اینکه شامل دستوراتی برای برنامه‌ها است، در عین حال دستورات خاصی را به ماشین می‌دهد که عملیات پردازشی را کنترل کند. به عنوان مثال در کارت‌هایی که نشان داده شده است می‌بینید که در بخشی از کارت‌ها یک برنامه به زبان فورترن وجود دارد و در بخشی دیگر، داده‌ای برای برنامه وجود دارد، اما دستوراتی مانند دستور Load هم وجود دارد که مشخص می‌کند،

عملیات کاری چگونه خواهد بود؟ این دستور نشان می دهد که برنامه بعد از اینکه وارد سیستم شد باید آماده اجرا شود. دستور Run اجرا را نشان می دهد و به نوعی یک مدل دسته ای از کارها را نشان خواهد داد.

دیدگاه دیگری که در تاریخچه سیستم های عامل مطرح بود و در نسل سوم کامپیوترها مطرح شد بحث های multi programming (سیستم های چند برنامه گی) بود، هدف سیستم های چند برنامه گی این بود که چند برنامه به طور همزمان در حافظه حضور داشته باشند.



به عنوان مثال در شکل می بینید که در یک لحظه سه کار به طور همزمان در حافظه قرار داده شده است و در صورت وجود یک پردازشگر هیچوقت این سه کار نمی توانند همزمان اجرا شوند، بنابراین باید از تکنیک هایی استفاده می شد که بتوانند اجرای همزمان این کارها را پشتیبانی کنند. یکی از این تکنیک ها CPU scheduling یا زمان بندی CPU یا بحث Time sharing است.

از انواع سیستم های چند برنامه گی به یک سیستم بسیار معروف چند برنامه گی به نام Simple multiprogramming یا چند برنامه گی ساده می توان اشاره کرد.

یک خاصیت مهم سیستم‌های چند برنامه‌گی ساده آن است که در این سیستم‌ها تنها دو Job همزمان در سیستم حضور داشت نوع این Job ها با یکدیگر متفاوت بودند، یک Job را به عنوان Cpu Job یا limited یا cpu bounded یا کارهای محدود به CPU می‌شناختند و دیگری، کارهای محدود به IO یا IO limited مطرح می‌شد.

کارهای محدود به IO کارهایی بودند که مدت زمان پردازش آنها زیاد نبود، اما مدت زمان بسیار زیادی طول می‌کشید که داده‌های مربوط به پردازش وارد سیستم شود. مثال‌های ساده‌ای می‌توان از کارهای IO limited ذکر کرد. به عنوان مثال فرض کنید، می‌خواهیم میانگین سنی داده‌های کل افراد موجود در یک جامعه ۲۰ میلیون نفری را پیدا کنیم، مدت زمان زیادی طول می‌کشد که اطلاعات مربوط به افراد وارد سیستم شود اما محاسبه میانگین سنی، زمان کوتاهی را صرف خواهد کرد.

CPU limited نوع دیگر کارهایی است که در سیستم‌های چند برنامه‌گی ساده وجود داشت. که از جمله کارهایی هستند که زمان پردازش زیادی را صرف خواهند کرد. از جمله این کارها حل مسائل کلاسیک است که معمولاً زمان زیادی برای حل مسأله صرف خواهد شد. اما تعریف مسأله و ورود اطلاعات مدت زمان زیادی طول نخواهد کشید، در یک سیستم چند برنامه‌گی ساده این دو نوع کار با یکدیگر چند برنامه می‌شوند یعنی همزمان در سیستم قرار می‌گیرند، حال برای اینکه بتوان کنترل این کارها را انجام داد و زمان بندی و نحوه چند برنامه شدن آنها را بدست آورد از یک پروتکل ساده استفاده می‌شود. به این صورت که اگر کار IO limited نیاز به CPU داشته باشد، CPU بلافاصله در اختیار او قرار می‌گیرد با این فرض که ما می‌دانیم که کار IO limited مدت زمان زیادی CPU را نیاز ندارد، پس بلافاصله CPU به وی تخصیص داده می‌شود. تا عملیات آن، انجام شده و به پایان برسد و کار CPU Limited بتواند با فراق بال بیشتری عملیات پردازشی خود را دنبال کند، دومین پروتکل به این گونه تعریف می‌شود که اگر کار CPU limited عملیات IO نیاز داشته باشد، دستگاه‌های IO بلافاصله در اختیار وی قرار می‌گیرد. در اینجا نیز استدلالی مشابه مرحله اول می‌توان داشت. در هر حال یکی از نمونه‌های سیستم‌های چند برنامه‌گی سیستم چند برنامه‌گی ساده بود که در نسل سوم کامپیوترها از آن استفاده می‌شد.

انواع سیستم های عامل

در این مبحث دیدگاه‌های مختلف دنیای سیستم‌های عامل یا مجموعه سیستم‌های عامل یا انواع سیستم‌های عاملی را که وجود دارند مطرح خواهد شد و یک تعریف کلی از هر کدام از این سیستم‌های عامل ارائه می‌شود.

اولین نوع سیستم‌های عامل، سیستم‌های عامل کامپیوترهای **main frame** است: کامپیوترهای بسیار بزرگی که می‌توانستند تعداد زیادی کاربر را به طور همزمان مدیریت کنند و کاربران مختلف از طریق ترمینال‌های مختلف به این ماشین‌ها وصل می‌شدند و سیستم‌های عامل آنها وظیفه تعیین هویت کاربران و سپس در اختیار قرار دادن منابع به کاربران را بر عهده داشتند.

دومین نوع سیستم‌های عامل، سیستم‌های عامل سرور یا سیستم‌های عامل کارگزار هستند که وظیفه آنها نصب بر روی ماشین‌های سرور و انجام خدماتی است که ماشین‌های **Client** نیاز به این خدمات دارند. از انواع دیگر سیستم‌های عامل، سیستم‌های عامل **multi processor** یا چند پردازنده می‌باشد که وظیفه عمده آنها مدیریت دقیق چندین پردازنده به طور همزمان است.

در مباحث بعدی عنوان خواهد شد که تفاوت معنایی بین سیستم‌های عامل چند پردازنده و سیستم‌های عامل **multi computer** وجود خواهد داشت. سیستم‌های عامل **multi computer** را در لیست سیستم‌های عامل مطرح نکردیم چون در گروه مباحث مربوط به شبکه قرار می‌گیرند. اما در بحث سیستم‌های توزیع شده **multi computer** را هم مورد بررسی قرار خواهیم داد.

نوع دیگری از این دنیای سیستم‌های عامل، سیستم‌های عامل **personal camputer** ها (pcها) هستند که به هر حال کاربران آنها نیازهای متفاوتی دارند و باید امکانات متفاوتی را در اختیار کاربران قرار دهند.

از دیدگاه دیگری نیز می‌توان سیستم‌های عامل را دسته بندی کرد:

سیستم‌های عامل **real time** یا بلادرنگ، سیستم‌هایی هستند که به گونه‌ای باید طراحی شوند که **task** ها و وظایف تعریف شده در این سیستم‌ها بتوانند در یک بازه زمانی مشخص که غالباً این بازه زمانی را خط مرگ (**Dead line**) می‌نامیم انجام دهند و چنانچه برخی از فعالیت‌ها در زمان مشخص انجام نشود

آنگاه می توان گفت که انجام آنها ضرورتی ندارد و در اصطلاح این **task** ها و **job** ها از سیستم حذف می شوند.

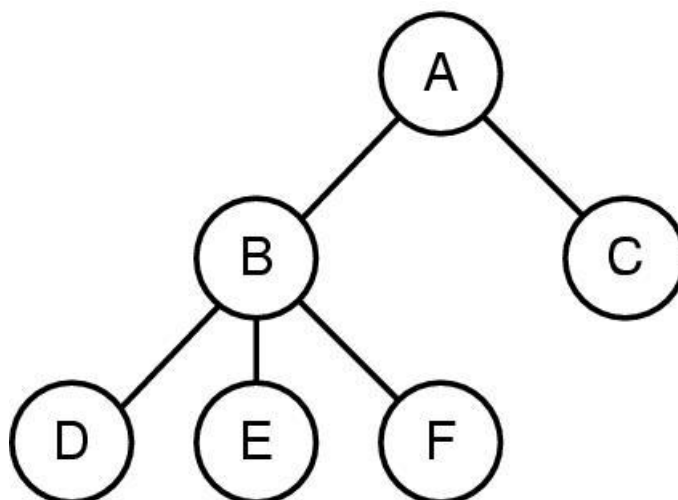
سیستم های عامل **embedded** یا سیستم های عامل نهفته، سیستم های عاملی هستند که معمولاً در دل سیستم دیگری جای می گیرند و کنترل اجزا را به صورت اشتراکی به کمک سیستم دیگری انجام خواهند داد.

آخرین گروه سیستم های عامل، **Smart cart** ها یا کارت های هوشمند هستند که نیازمند سیستم های عامل خاص خود هستند. موارد ذکر شده، نمونه هایی از انواع سیستم های عامل بود و نمی توان مدعی بود که سیستم های عامل دیگری وجود ندارد. از جمله مواردی که شاید این روزها بیشتر مورد توجه عموم قرار می گیرد سیستم های عامل گوشی های موبایل هستند که به طور خاص نمونه ای از این سیستم های عامل، سیستم های **Symbian** هستند.

پردازش، فرایند، فرا روند، (process):

فرایند یک برنامه در حال اجراست، به عبارت دیگر برنامه نوشته شده ای که توسط کاربر یا توسط سیستم عامل اجرا می شود را یک فرایند می نامیم. تعاریف دیگری نیز برای فرایند وجود دارد. برای مطالعه بیشتر می توانید به کتاب های درس سیستم عامل ۱ مراجعه کنید. پس فرایند یک برنامه در حال اجراست، این برنامه در حال اجرا برای اینکه بتواند اطلاعات مربوط به خود را داشته باشد باید تعامل مستقیم با سیستم عامل داشته باشد. به عبارت دیگر سیستم عامل یک فرایند را اجرا می کند و اطلاعات مربوط به فرایند را در قسمتی از حافظه اصلی قرار می دهد که این قسمت از حافظه اصلی را معمولاً **PCB** یا **Process control block** می نامند. اطلاعات جامعی در مورد **Process** در **PCB** وجود دارد. از جمله این اطلاعات می توان به **Process ID** یعنی شماره آن که معمولاً شماره ای یکتا در میان کلیه فرایندهای موجود در سیستم است اشاره کرد. دیگر اطلاعات موجود در **PCB** عبارتند از زمان اجرای فرایند تا لحظه مورد نظر، منابع در اختیار وی و حتی در هنگام تعویض متن، اطلاعات مربوط به ثبات های **CPU**. اما در یک فرایند معمولاً یک **Process tree** یا درخت فرایند نیز وجود دارد. وجود یک **Process tree** در یک سیستم عامل معمولاً به معنای این است که سیستم عامل اجازه می دهد فرایندهای موجود در سیستم تولیدمثل کنند، یعنی یک فرایند می تواند به یک یا چند فرایند دیگر تبدیل شود. از جمله سیستم های عامل

بسیار معروفی که امکان ایجاد فرآیندها و تولیدنسل فرآیندها را به سادگی فراهم کرده است سیستم عامل Unix است.



به عنوان مثال چنانچه به شکل دقت نمائید خواهیم دید که فرآیند A دو فرآیند فرزند، بنام فرآیندهای B و C ایجاد کرده است، ایجاد دو فرآیند B و C و حضور آنها در سیستم می تواند تحت تأثیر فرآیند A قرار گیرد، به این معنی که ممکن است فرآیندهای B و C فرآیندهای مستقلی از A باشند یعنی اگر فرآیند A به پایان برسد فرآیندهای B و C کماکان به زندگی خود ادامه دهند، اما در دیدگاه دیگری فرض می شود که فرآیندهایی که از یک فرآیند پدر به وجود آمده اند با مرگ فرآیند پدر حتماً از بین می روند. اعمال هر دوی این استراتژیها در سیستم امکان پذیر است. با توجه به شکل، فرآیند B نیز سه فرآیند دیگر یعنی فرآیندهای F و E و D را تولید کرده است، همین داستان به صورت سلسله مراتبی برای این فرآیندها نیز قابل تامل است.