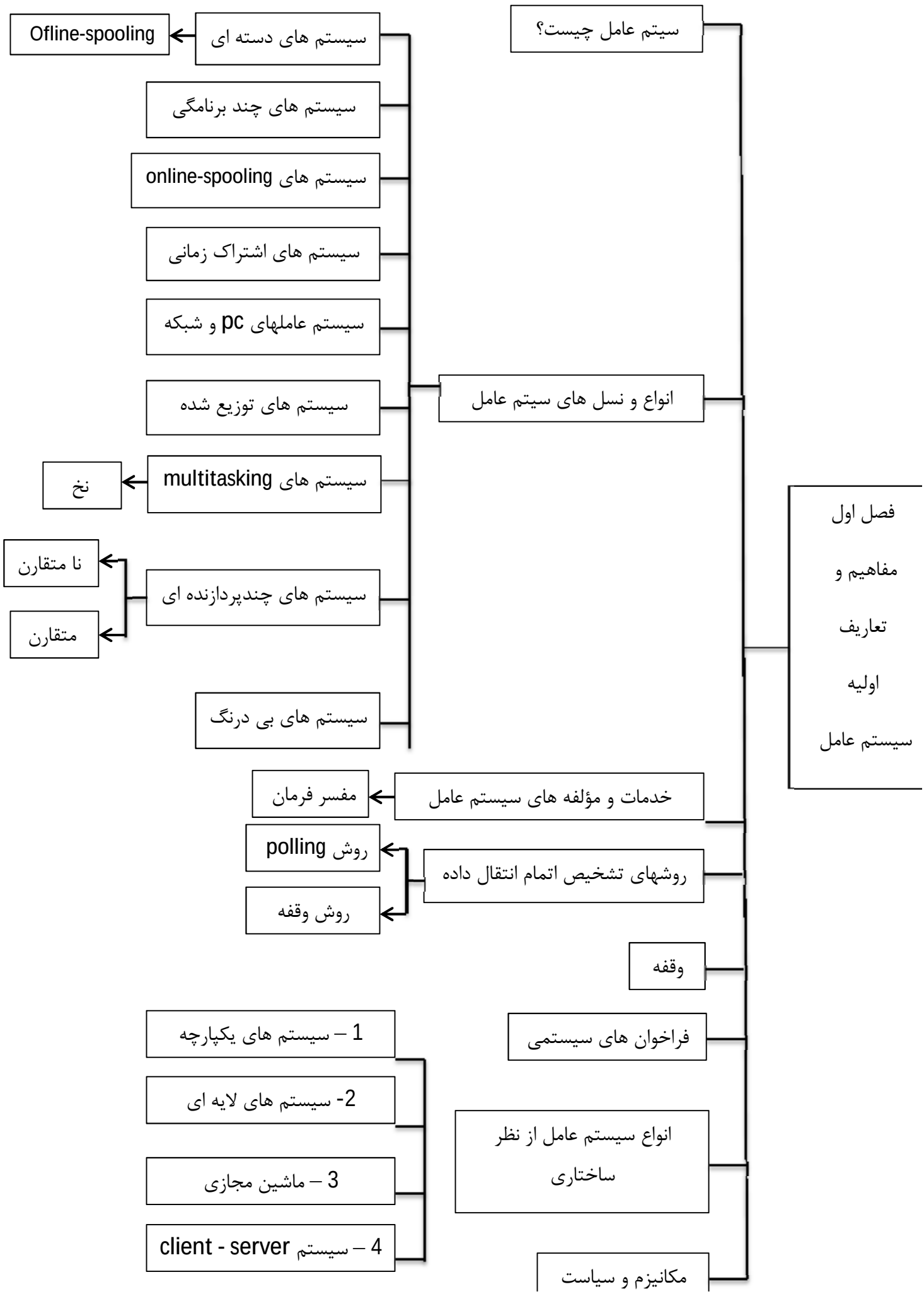
جزوه درس سیستم عامل

مدرس: نوید محسنی

مقطع کاردانی

دانشکده فنی امام محمدباقر(ع) ساری


فصل اول
مفاهیم و
تعاریف
اولیه
سیستم عامل
سیتم عامل چیست؟
انواع و نسل های سیتم عامل
سیستم های دسته ای
Ofline-spooling
سیستم های چند برنامگی
سیستم های online-spooling
سیستم های اشتراک زمانی
سیستم عاملهای pc و شبکه
سیستم های توزیع شده
سیستم های multitasking
نخ
سیستم های چندپردازنده ای
نا متقارن
متقارن
سیستم های بی درنگ
خدمات و مؤلفه های سیستم عامل
مفسر فرمان
روشهای تشخیص اتمام انتقال داده
روش polling
روش وقفه
وقفه
فراخوان های سیستمی
انواع سیستم عامل از نظر ساختاری
1 – سیستم های یکپارچه
2- سیستم های لایه ای
3 – ماشین مجازی
4 – سیستم client - server
مکانیزم و سیاست

فصل اول

مفاهیم وتعاریف اولیه سیستم عامل

اجزای مختلف کامپیوتر

همانطور که می دانید کامپیوتر از دو بخش کلی سخت افزار و نرم افزار تشکیل شده است. به طور دقیق تر می توان اجزای یک سیستم کامپیوتری را به صورت زیر ترسیم کرد.

	برنامه های کاربردی (حسابداری ، مرورگر وب و ...)
	مفرپسر فرمان ، مترجم ها ، ادیتور ها
برنامه های سیستمی	سیستم عامل
	زبان ماشین
	ریز برنامه نویسی
	دستگاه های فیزیکی

دربسیاری از سیستم ها هنگام اجرای دستورات زبان ماشین ، ابتدا این دستورات توسط یک ROM به یک سری دستورالعمل های ابتدایی تر به نام ریز دستور یا میکرو کد تبدیل شده و سپس توسط سخت افزار اجرا می گردند که به این ماشین ها CISC (Complex Instruction Set Computer) گفته می شود.

در بعضی از کامپیوتر ها به نام RISC (Reduced Instruction Set Computer) این سطح ریزبرنامه نویسی وجود نداشته و دستورات زبان ماشین مستقیما توسط سخت افزار اجرا می گردند. تعداد دستورات ماشین های RISC کمتر از کامپیوتر های CISC می باشد.

سیستم عامل چیست؟

به طور کلی نرم افزار های کامپیوتر به دو گروه تقسیم می شوند : یکی برنامه های سیستمی که عملیات کامپیوتر را مدیریت می کنند و دیگری برنامه های کاربردی. سیستم عامل (Operating System = OS) اصلی ترین برنامه ی سیستمی است که به عنوان رابط بین کاربر و سخت افزار عمل می کند.

سیستم عامل دو وظیفه (یا هدف) اصلی دارد :

الف) سیستم عامل استفاده از کامپیوتر را آسان می سازد. این بدان معناست که مثلاً کاربر یا برنامه نویس بدون درگیر شدن با مسائل سخت افزاری دیسکها به راحتی فایلی را بر روی دیسک ذخیره و حذف کند. این کار در واقع با به کار بردن دستورات ساده ای که فراخوان های سیستمی (System Calls) را صدا می زنند انجام می پذیرد.

در صورت عدم وجود سیستم عامل کاربر و یا برنامه نویس می بایست آشنایی کاملی باسخت افزار های مختلف کامپیوتر (مثل مونیتور، فلاپی ، کیبورد ، وغیره)داشته باشد و روتین هایی برای خواندن و نوشتن آنها به زبان های سطح پایین بنویسد. از این جنبه به سیستم عامل با عنوان ماشین توسعه یافته (Extended machine) یا ماشین مجازی (Virtual machine) یاد می شود که در واقعیت سخت افزار را از دید برنامه نویسان مخفی می سازد.
ب) وظیفه دوم سیستم عامل مدیریت منابع (Resource Management) می باشد , یعنی سیستم عامل باعث استفاده بهینه و سودمند (اقتصادی) از منابع فیزیکی و منطقی سیستم می گردد. منظور از منابع فیزیکی پردازنده ها, حافظه ها , دیسکها ، ماوس ها , چاپگرها ، پورتها و غیره و منظور از منابع منطقی اطلاعات ، فایلها و غیره هستند.یک سیستم کامپیوتری منابع نرم افزاری و سخت افزاری بسیاری دارد که ممکن است در حین اجراء برنامه لازم باشند، سیستم عامل همانند مدیر منابع عمل کرده و آنها را برحسب نیاز به برنامه های مشخصی تخصیص می دهد مثلا اگر دو برنامه همزمان بخواهند ازچاپگر استفاده کنند، سیستم‌عامل اطلاعات خروجی آنها را در بافری ریخته و سپس به ترتیب و بر اساس اولویتی از پیش تعیین شده اطلاعات را به چاپگر می فرستد . بنابراین کار سیستم عامل این است که بداند چه کسانی از کدام منابع استفاده می کنند، به درخواستهای جهت دریافت منابع رسیدگی کند ، حسابداری استفاده از منابع را نگهداری کند و در برابر تداخل درخواست های مختلف میانجیگری کند. پس با نگرش از بالا به پایین سیستم عامل فراهم کننده واسطه ای راحت برای کاربران است و با نگرش از پایین به بالا سیستم عامل مدیر کلیه اجزاء و منابع سیستم می باشد. به بیانی می‌توان گفت هدف اصلی سیستم عامل استفاده بهینه و حداکثر از ماشین و وظیفه سیستم عامل کنترل تمامی منابع و به تعادل رساندن درخواستها و منابع موجود است.

با توجه به توضیحات فوق می‌توان گفت سیستم عامل مشابه دولت است یعنی مانند دولت سیستم عامل محیطی را که درون آن سایر برنامه ها بتوانند کار مفید انجام دهند، ایجاد می کند و هسته سیستم عامل به خودی خود کاری مفید و نهایی را برای کاربر انجام نمی دهد. به عبارت دیگر سیستم عامل برنامه کنترلی است که اجرای برنامه‌های کاربر و استفاده از سخت افزار سیستم را کنترل می کند.

سیستم عامل معمولا اولین برنامه ای است که پس از بوت شدن در حافظه باز می شود. پس از باز شدن ، قسمتی از سیستم عامل بطور دائم در حافظه باقی (Resident) می‌ماند. قسمتهای دیگر با توجه به کاربرد کامپیوتر توسط کاربر از دیسک به حافظه آورده می شود. به قسمت اصلی سیستم عامل که وظایف مهم آن را انجام می دهد هسته یا kernel گفته می شود. هسته سیستم عامل برنامه ای است که در تمامی اوقات بر روی کامپیوتر در حال اجراست. سیستم عامل و معماری کامپیوتر اثر زیادی بر روی یکدیگر داشته اند. یعنی جهت سهولت کار با سخت افزارهای

جدید ، سیستم‌عامل ها توسعه یافتند همچنین در اثنای طراحی سیستم عامل ها ، مشخص شد که تغییراتی در طراحی سخت افزار می تواند سیستم عاملها را ساده تر و کار آمد تر سازد. هر چند که تطبیق نسلهای کامپیوتر با نسلهای سیستم عامل کار درستی نیست ولی این تطبیق که در ادامه انجام می دهیم علت ایجادسیستم عامل های جدید را مشخص می سازد. در نسل اول کامپیوترها (55 – 1945) که از لامپ خلأ برای ساخت آنها استفاده می شد ، زبان های برنامه نویسی (حتی اسمبلی) ابداع نشده بودند و سیستم عامل نیز اصلا وجود نداشت . روند کار به این صورت بود که برنامه نویسان تنها در یک فاصله زمانی مشخص حق استفاده از کامپیوتر بزرگ و گران قیمت را داشتند. آنها برنامه های خود را توسط تخته مدار سوراخدار (و بعدها توسط کارتهای پانچ) وبه زبان ماشین به کامپیوتر می دادند. اکثر برنامه های محاسبات عددی معمولی مانند جداول سینوس و کسینوس بود.

سیستم های دسته ای Batch System

در نسل دوم ، کامپیوتر ها (65 - 1955) از ترانزیستور ساخته شدند. طریقه کار با این کامپیوتر های نسل دوم از طریق یک کنسول (Console) بود که تنها اپراتور مخصوص کامپیوتر با آن کار می کرد و کاربران به طور مستقیم با این کاربران محاوره (Interaction) نداشتند. کاربر ابتدا برنامه خود را به زبان فرترن با اسمبلی بر روی کاغذ می نوشت سپس توسط دستگاه Card punch برنامه را روی کارت های سوراخدار منتقل می ساخت.بعد از این دسته کارت تهیه شده که شامل برنامه ، داده و کارتهای منترل بود به صورت یک کار (Job) تحویل اپراتور داده می شد. اپراتور بعد از اتمام کار قبلی ، دسته کارت جدید را به کامپیوتر می داد تا برنامه را اجرا کند در انتها خروجی برنامه (که غالبا چاپی بود) را به کاربر تحویل می داد. سیستم عامل در این کامپیوتر های اولیه ساده بود و وظیفه اصلی آن انتقال کنترل اتوماتیک از یک کار به کار دیگری بود. سیستم عامل همواره مقیم در حافظه بود و در هر لحظه فقط یک برنامه اجرا می شد.هنگامی که اپراتور مشغول گذاشتن نوار ها یا برداشتن کاغذ های چاپ شده بود وقت زیادی از این کامپیوتر های گران قیمت به هدر می رفت. برای رفع مشکل فوق سیستم های دسته ای (Batch System)ابداع شد. یعنی ابتدا یک سبد پر از دسته کارتها در اتاق ورودی جمع آوری می شد، کلیه آنها به وسیله دستگاه کارتخوان یک کامپیوتر کوچک نسبتا ارزان (مثل IBM 1401) خوانده شده وبر روی یک نوار ذخیره می گردید. سپس اپراتور نوار را برداشته بر روی کامپیوتر اصلی و گران قیمت که محاسبات را انجام می داد (مثل IBM 7094) نصب می کرد. بعد از آن برنامه ای را اجرا می کرد(یعنی سیستم عامل) تا اولین کار را از روی نوار برداشته و اجرا کند. خروجی بر روی نوار دیگری نوشته می شد. پس از اتمام هر کار سیستم عامل به صورت خودکار کار بعدی را از نوار می خواند. پس از اجرا همه برنامه ها، اپراتور نوار خروجی را برداشته و دوباره روی کامپیوتر IBM 1401 منتقل می ساخت تا عملیات چاپ خروجی ها به صورت offline انجام شود. به این روش کار offline spooling نیز گفته می شود. بیشتر برنامه های نسل دوم به زبان فرترن یا اسمبلی برای محاسبات مهندسی و علمی مثل مشتقات جزئی به کار می رفت.

یکی از معایب ofline - spooling زیاد بودن زمان برگشت (گردش) (turnaround time) است ، یعنی تأخیر زمانی مابین تحویل کار و تکمیل کار و همچنین در این سیستم اولویت بندی به معنای واقعی وجود ندارد. تنها روش بدست آوردن اولویت این بود که نوار کارهای مهم را ابتدا در ماشین اصلی قرار دهند. حتی در اینصورت هم باید چندین ساعت صبر می کردند تا خروجی ها ظاهر شوند. همچنین نیاز به سخت افزار اضافی (مثل کامپیوتر های 1401) از دیگر معایب این روش بود. پس معایب این روش عبارت است از : 1- گردش طولانی تر کار 2- عدم وجود اولویت 3 نیاز به سخت افزار اضافی.

از مزایای سیستم offline spooling نسبت به سیستم های قبل از آن می توان موارد زیر را ذکر کرد : 1- راندمان بهتر 2- عملیات ساده تر 3- سهولت برای استفاده از راه دور

تذکر: در سیستم های اولیه CPU گرانترین جزء کامپیوتر بود و بدین جهت نیاز به بالا بردن درصد استفاده از CPU فاکتور اصلی در طراحی سیستم عامل ها بود.

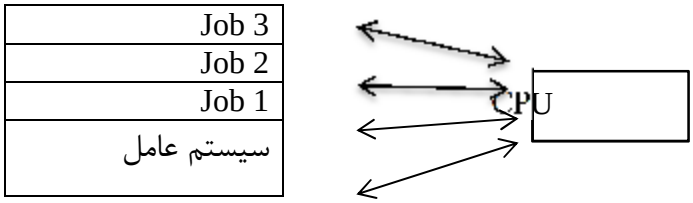
سیستم ها چند برنامگی Multiprogramming

در نسل سوم کامپیوتر ها (80- 1965) از مدارات مجتمع (Integrated Circuit = IC) برای ساخت کامپیوتر ها استفاده شد. به طور کلی برنامه هارا می توان به دو دسته تقسیم کرد : یکی برنامه ها با تنگنای محاسباتی CPU (CPU Limited یا bound) مانند محاسبات علمی سنگین که بیشتر زمان کامپیوتر صرف محاسبات CPU می شود و دیگری برنامه ها با تنگنای I/O (I/O Limited) مانند برنامه های تجاری که بیشتر زمان کامپیوتر صرف ورود داده ها و خروج اطلاعات می شود.

یک اشکال مهم سیستم های دسته ای این است که وقتی کار جاری برای تکمیل یک عملیات I/O مثلا بر روی نوار گردان منتظر می شود، در این حال CPU بیکار می ماند و مجبور است صبر کند تا عملیات I/O به اتمام برسد. در برنامه های CPU Limited این اتلاف وقت اندک است ولی در برنامه های I/O Limited ممکن است حدود 80 تا 90 درصد وقت CPU به هدر برود.

برای رفع این مشکل از تکنیک Multiprogramming استفاده می شود. بدین ترتیب که حافظه به چند قسمت تقسیم شده و در هر قسمت یک برنامه مجزا قرار داده می شود. وقتی که یک کار برای تکمیل عملیات I/O منتظر می ماند ، پردازنده به کار دیگری داده می شود. اگر تعداد کارهای موجود در حافظه کافی باشد می توان CPU را تقریبا صد در صد مشغول نگه داشت. البته نگهداری همزمان چند برنامه در حافظه نیاز به مدیریت خاص حافظه دارد تا برنامه ها بر همدیگر اثر سوء نداشته باشند. لذا مدیریت حافظه بحث مهمی در سیستم عامل می باشد(شکل زیر).

حافظه



پس سیستم های چند برنامگی فقط یک پردازنده دارند و به کمک مکانیزم وقفه بین کارهای CPU Limited و I/O Limited سوئیچ می شود و به ظاهر اجرای این برنامه ها به صورت موازی وهمزمان صورت می گیرد و بدین ترتیب بهره وری از سیستم افزایش می یابد.

یک شیوه برای بالا بردن راندمان CPU آن است که تعدادی کار CPU – Limited را با I/O – Limited مخلوط کنیم ولی در عمل امکان تقسیم بندی کارها قبل از اجرا شدنشان وجود ندارد . از طرف دیگر برنامه ای ممکن است در ابتدا CPU – Limited باشد ولی در حین اجرا تبدیل به I/O-Limited گردد.

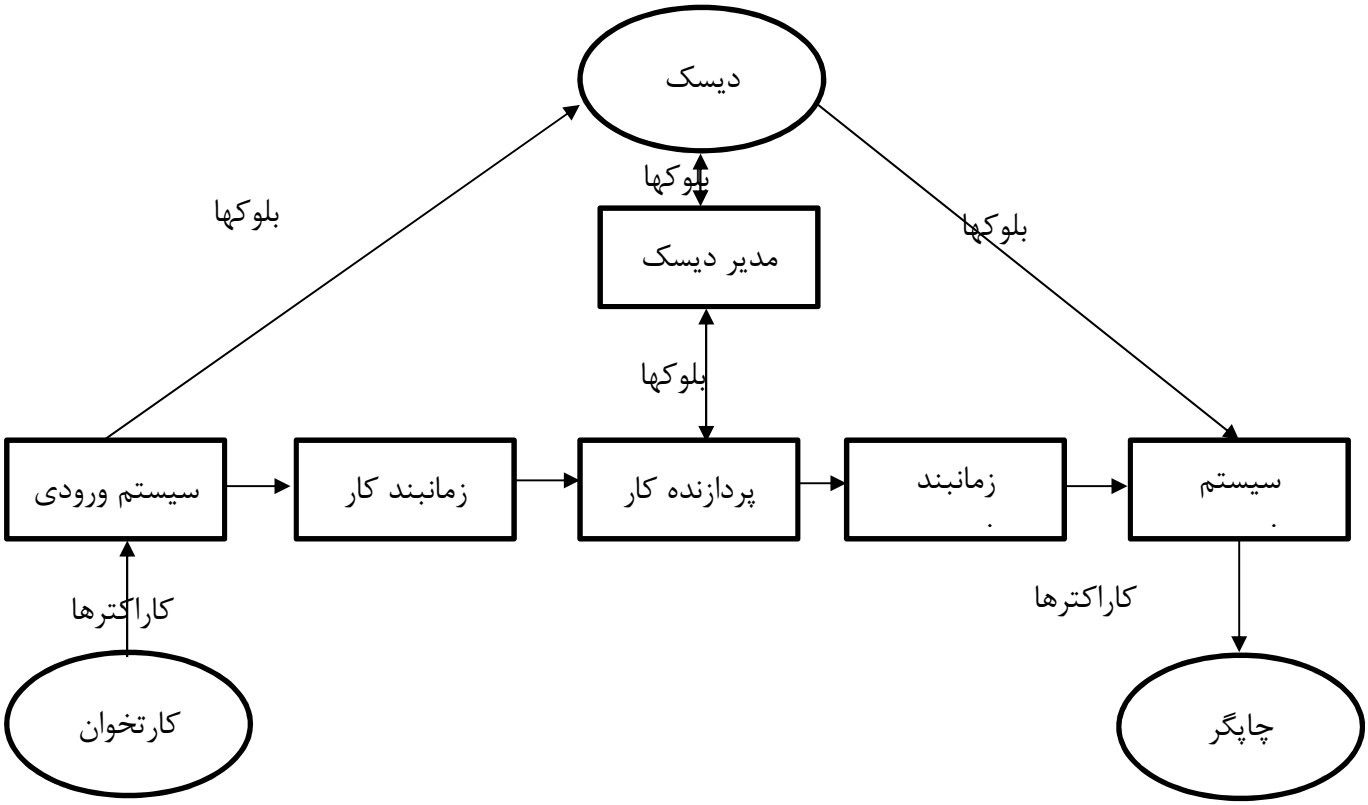
سیستم های Spooling

یکی دیگر از ویژگی های سیستم های نسل سوم Spooling (یا OnLine Spooling) است که معمولا همراه چند برنامگی استفاده می شود . این کلمه مخفف عبارات (Simultancous Peripheral Operation OnLine) می باشد.

در این سیستم به جای آنکه کارتها از دستگاه کارت خوان مستقیماً وارد حافظه گردند و توسط CPU پردازش شوند ابتدا کاراکتر به کاراکتر در بافری در حافظه قرار گرفته و سپس به صورت بلوکی بر روی دیسک نوشته می شوند. وقتی که برنامه کاربر اجرا می شود و از سیستم عامل تقاضای ورودی می کند ، اطلاعات ورودی به صورت بلوکی و با سرعت زیاد از دیسک خوانده می شوند. به طور مشابه هنگامی که برنامه برای خروجی چاپگر را احضار می کند، خط خروجی در یک بافر کپی شده و سپس در دیسک نوشته می شود. پس اطلاعات خروجی از دیسک بر اساس ترتیب و اولویت در چاپگر چاپ می شوند. در واقع اسپولینگ عمل I/O یک کار را با عمل محاسباتی کار دیگر روی هم می اندازد(overlap). در سیستم اسپولینگ در حالیکه ورودی یک کار از دستگاه ورودی خوانده می شود ، کار دیگری در حال چاپ شدن است. در همین بین حتی کار دیگری میتواند در حال پردازش و اجرا باشد . در اسپولینگ برنامه عملیات ورودی وخروجی اش را متناسب با سرعت دیسک (که سریع است) انجام می دهد و نه متناسب با سرعت کارتخوان یا چاپگر (که خیلی کند هستند). بنابراین سیستم مذکور باعث استفاده بهینه از CPU و وسایل I/O می شود و سرعت عمل را بالا می برد. در این سیستم دیگر نیازی به کامپیوتر های 1401 ، نوار گردانهای اضافی و حمل نوار ها (مانند سیستم های دسته ای) نداریم. با توجه به توضیحات فوق مزایای سیستم اسپولینگ (نسبت به سیستم

دسته ای) عبارتند از : 1- بالابودن راندمان CPU و وسایل I/O 2- گردش سریعتر کار (turnaround time) در اینجا برعکس سیستم های دسته ای لازم نیست برای کامل شدن نوار های ورودی یا خروجی صبر کرد 3- دسترسی با اولویت. هنگامی که اطلاعات مستقیماً از طریق کارتخوان یا نوار وارد می شوند چون دسترسی به صورت سریال است امکان ندارد که اجرای آنها با ترتیب متفاوتی انجام گیرد، یعنی کارها الزاماً به ترتیب ذخیره شدن روی نوار اجرا می شوند. اما با استفاده از دیسک گه رسانه ای با دسترسی تصادفی و مستقیم (direct access) است می توان بین کار ها زمان بندی کرده و کار با اولویت را ابتدا اجرا کرد هر چند که قبل از همه وارد نشده باشد. 4- می‌توان هم‌زمان چند مدرک ورودی یا خروجی داشت چون ورودی‌ها و خروجی‌ها در دیسک ذخیره می‌شوند.

بلوک دیاگرام یک سیستم اسپولینگ را می‌توان به صورت زیر ترسیم کرد:



1- **سیستم ورودی**: کاراکترهایی که توسط کارخانه وارد می‌شوند را در بلوک‌هایی جمع آوری کرده و به کمک مدیر دیسک این بلوک ها را بر روی دیسک می‌نویسد. در انتهای هر مدرک ورودی اطلاعاتی راجع به آن مدرک (مانند محل آن بر روی دیسک، اولویت، اسم استفاده کننده) به قسمت زمان‌بندی کار و فرستاده می‌شود.

2- **زمان بند کار (JS – Job Scheduler)**: این زمان بند یک 25 کارهای موجود در ماشین و اطلاعات لازم در مورد مدارک ورودی مورد نیاز هر یک را نگه می‌دارد. به این نیست انبار کار یا Jobpool یا Joblist نیز گفته می‌شود. زمانبند کار به پردازنده کار می‌گوید که کدام کار بدی را اجرا کند. برای این منظور اطلاعاتی در مورد محل کار و

مدارک ورودی آن بر روی دیسک را به پردازنده کار می‌دهد. همچنین اگر کارهای متعددی منتظر ورود به حافظ باشند و فضای کافی برای همگی در دسترس نباشد، زمانبند کار تعدادی از آنها را انتخاب کرده و به حافظه می‌آورد.

3- **پردازنده کار (Job processor)**: کار داده شده را اجراء می کند. این پردازنده محل کامپایلرها و سایر نرم افزاری های سیستم را بر روی دیسک می داند. هنگام اجراء پردازنده کار خروجی های خود را به صورت بلوکی بر روی دیسک می نویسد و مدارک خروجی را تشکیل می دهد پردازنده کار اطلاعاتی راجع به محل و اولویت مدارک خروجی به زمانبند خروجی می دهد.

4- **زمانبند خروجی (output scheduler)**: لیستی از مدارکی که باید چاپ شوند را نگه می دارد. وقتی که چاپگر آزاد شد، این زمابند مدارک بعدی را برای چاپ انتخاب کرده و مخل مدرک بر روی دیسک را به سیستم خروجی می گوید.

5- **سیستم خروجی**: بلاکهای خروجی را از روی دیسک خوانده و کاراکتر به کاراکتر (یا خط به خط) آنها را به چاپگر می فرستند.

6- **مدیر دیسک (Disk Manager)**: که وظایف خواندن و نوشتن یک بلاک بر دیسک، تخصیص یک بلاک خالی روی دیسک و برگرداندن یک بلاک به مجموعه فضای آزاد دیسک را بر عهده دارد درخواستهای مربوطه به دیسک در یک صف به نام DIQ (Disk Transfer Queue) ذخیره می گردد.

7- **هماهنگ کننده**: البته هر سیستم اسپولینگ یک هماهنگ کننده (Coordinator) دارد که مسئول زمانبندی پردازش های سیستم و فراهم کردن عملیاتی که جهت همگام کردن بکار می آیند می باشد. (هماهنگ کننده در شکل نشان داده نشده است). این عملیات توسط دو روال انجام می پذیرد:

Wait: پردازش جاری را متوقف کرده و دوباره وارد زمانبند می گردد.

Free: یک پردازش ویژه را جهت زمانبندی، آماده می کند.

اعمال Wait و free معادل دستورات sleep و wake up فرآیند هاست که در فصل بعدی شرح می دهیم.

نکته: بافر کردن امکان می دهد که عمل I/O یک کار با عمل پردازش همان کار همزمان گردد در حالیکه Spooling امکان می دهد عملیات I/O و پردازش چندین کار باهم همزمان گردند.

در چند برنامگی اجرای یک برنامه تا هنگام عملیات I/O ادامه پیدا می کند، سپس عمل I/O آن شروع شده و همزمان CPU اجرای برنامه دیگری را آغاز می کند. ولی در Spooling می توان چند کار را همزمان اجرا کرد.

تذکر: در سیستم های On line پردازنده مستقیماً با دستگاه های I/O در ارتباط است ولی در سیستم های Off line یا ارتباط غیر مستقیم، پردازنده با دستگاه های I/O به طور مستقیم در ارتباط نیست.

سیستم های اشتراک زمانی Time-Sharing

این سیستم ها از اوایل سالهای 1970 در نسل سوم کامپیوترها معمول شدند. سیستم اشتراک زمانی در واقع تعمیم سیستم چند برنامگی است.

در سیستم های چند برنامگی کاربر ارتباطی با کامپیوتر نداشت و خطایابی برنامه ها مشکل بود چرا که زمان برگشت نسبتاً طولانی اجازه آزمایش کردنهای متعدد را نمی داد. در سیستم اشتراک زمانی کاربر به کمک ترمینال (Terminal) که شامل کی بودر (برای ورودی) و مونیتور (برای خروجی) است با کامپیوتر به صورت محاوره ای (interactive) رابطه برقرار می سازد. کاربر مستقیماً دستوراتی را وارد کرده و پاسخ سریع آن را روی مونیتور دریافت می کند. در این سیستم ها چندین کاربر به کمک ترمینالهایی که به کامپیوتر وصل است همزمان می توانند از آن استفاده کنند. در سیستم اشتراک زمانی فقط یک پردازنده وجود دارد که توسط مکانیزمهای زمانبندی بین برنامه های مختلف کاربرها با سرعت زیاد (مثلاً در حد میلی ثانیه) سوئیچ می شود و بنابر این هر کاربر تصور می کند کل کامپیوتر در اختیار اوست در اینجا تأکید بر روی میزان عملکرد کاربر است یعنی هدف فراهم کردن وسایل مناسب برای تولید ساده نرم افزار و راحتی کاربر می باشد و نه بالابردن میزان کاربرد منابع ماشین، کاربر می تواند در هر زمان دلخواه برنامه خود را آغاز یا متوقف سازد و یا برنامه را به صورت قدم به قدم اجراء و اشکال زدایی (debug) کند. سیستم های دسته ای برای اجرای برنامه های بزرگ که نیاز محاوره ای کمی دارند مناسب است ولی سیستم های اشتراک زمانی برای مواردی که زمان پاسخ کوتاه لازم است استفاده می شوند. در زمانی که کاربری در حال تایپ برنامه اش یا فکر کردن روی خطاهای برنامه اس می باشد CPU به برنامه کاربر دیگری اختصاص یافته تا آن را اجراء کند.

در سیستم های محاوره ای دستورات به دو صورت پیش زمینه (foreground) و پس زمینه (Background) اجرا می شوند، در نوع پیش زمینه با واردکردن دستور، تا هنگامی که دستور اجرا و تمام نشود اعلان سیستم ظاهر نمی گردد

ولی در نوع پس زمینه، پس از ورود دستور، بلافاصله اعلان سیستم ظاهر می شود تا دستور بعدی وارد گردد ولی در همین حین دستور قبلی در پشت زمینه در حال اجرا می باشد.

در این سیستم اشتراک زمانی هنگامی که چند کاربر همگی یک برنامه (مثل کامپایلر C) را احضار می کنند هر یک دارای کپی هایی از ثباتها، فضای داده‌ای، فضای پشته (stack) مخصوص به خود هستند ولی قسمت کد همگی یکسان و مشترک می باشد. در سیستم اشتراک زمانی وجود یک سیستم فایل ضروری است زیرا نمی توان در هر بار اجرای کار مدارک بزرگی را توسط ترمینالها وارد کامپیوتر کرد. لذا داده ها و برنامه های کاربر می بایست در دیسک ذخیره گردند. لذا بخش مدیریت فایلها یکی از بخش های اصلی سیتسم عامل می باشد. هر چند که می توان وظایف مدیر فایل (File Manager) و مدیر دیسک (Disk Manager) را در هم آمیخت ولی منطقاً این دو کاملاً متفاوت هستند. مدیر دیسک مسئول توزیع فضای خالی روی دیسک و نقل و انتقالات است، حال آنکه کارهای مدیر فایل بیشتر مربوط به مدیریت فهرستها، فایلها، امنیت فایلها و کارهای مربوط به حسابداری (Accounting) است. یکی دیگر از بخشهای مهم در سیستم اشتراک زمانی، مدیر ترمینال است که وظیفه تعیین هویت کاربران و پذیرش آنها را بر عهده دارد.

چند برنامگی و اشتراک زمانی مباحث اصلی سیستم عاملهای امروزی و در نتیجه مباحث اصلی این کتاب می باشند. اشتراک زمانی حالت ویژه ای از چند برنامگی است که در آن تعویض یک برنامه نه بر مبنای لحظه نیاز برنامه به عمل I/O بلکه بر مبنای یک برش زمانی انجام می پذیرد. از جمله سیستم عاملهای سنتی و مشهور که شامل این ویژگیها می باشد سیستم عامل UNIX است. مؤسسه IEEE یک استاندارد به نام POSIX برای UNIX بنا نهاد. POSIX حداقل واسط فراخوان سیستمی را تعریف می کند که سیستم های UNIX سازگار باید آن را پشتیبانی نمایند.

سیستم عاملهای کامپیوتر های شخصی و شبکه

سال 1980 تاکنون که مدارات مجتمع با مقیاس بزرگ LSI (Large Scale Integrated Circuit) ابداع شدند. به عنوان نسل چهارم کامپیوترها شناخته می شود. در این سالها کامپیوترهای شخصی با قیمتی ارزان و کارآئی بالا و محیط گرافیکی و محاوره ای بسیار خوب به سرعت گسترش یافتند. سیستم عاملهای اولیه بر روی PC ها (مانند DOS) فقط تک کاربره و تک برنامه ای بودند. ولی سیستم عاملهای امروزی آن مانند Windows NT خاصیتهای چند برنامگی، چند کاربره (Multiuser) و شبکه ای را دارا هستند. با توجه به هزینه اندک سخت افزار اهداف سیستم

عامل در طول زمان تغییر کرده است و برای PC ها به جای ماکزیمم کردن درصد استفاده CPU و وسایل جانبی، سیستم به سمت راحتی کاربر پیش می رود. به تدریج ویژگی های مهم سیستم عاملهای قدیمی در کامپیوتر های بزرگ (مانند حقاظت حافظه، حافظه مجازی، محافظت فایلها، همزمانی پرداشها و ...) بر روی سیستم های PC نیز پیاده سازی شده است.

هنگامی که کامپیوتر ها از طریق شبکه به هم وصل شوند به آنها ایستگاههای کاری (Workstation) می گویند. در یک سیستم عامل شبکه، کاربران از وجود ماشین های مختلف در شبکه باخبرند. آنها می توانند از دور وارد یک ماشین شوند و همچنین فایلهای یک ماشین را روی ماشین دیگر کپی کنند. هر کامپیوتر سیستم عامل محلی خودش را اجراء می کند و کاربر یا کاربران محلی مخصوص به خود را دارد.

سیستم های توزیع شده Distributed Systems

سیستم عامل توزیع شده در یک محیط شبکه ای اجراء می شود. در این سیستم قسمتهای مختلف برنامه کاربر بدون آنکه خود او متوجه شود می توانند همزمان در چند کامپیوتر مجزا اجراء شده و سپس نتایج نهائی به کامپیوتر اصلی کاربر برگردند. کاربران نباید از این موضوع باخبر شوند که برنامه آنها در کجا به اجراء در می آید. و یا فایلهای آنها در کجای شبکه قرار دارد و همه این کارها باید توسط سیستم عامل به صورت خودکار انجام گیرد. به عبارتی دیگر سیستم باید از دید کاربر شفاف باشد و هر چیز را با نام آن فراخوانی کند و کاری به آدرس آن نداشته باشد.

با توجه به توضیحات فوق سیستم عامل های توزیع شده به مراتب پیچیده تر از سیستم عاملهای شبکه هستند. یکی از مزایای مهم سیستم های توزیع شده سرعت بالای اجرای برنامه هاست چرا که یک برنامه همزمان می‌تواند از چندین کامپیوتر برای اجراء شدنش استفاده کند. همچنین به علت توزیع شدن اطلاعات، بانکهای اطلاعاتی حجیم می تواند روی یکسری کامپیوتر های شبکه شده قرار بگیرند و لازم نیست که همه اطلاعات به یک کامپیوتر مرکزی فرستاده شود (که در نتیجه این نقل و انتقالات حجیم زمان زیادی به هدر می رود.)

در شبکه هر کامپیوتر می تواند سیستم عامل ویژه خود را داشته باشد و تنها یک نرم افزار مدیریت شبکه به سیتسم عامل اضافه شده است ولی در سیستم عامل توزیع شده تنها یک سیستم عامل مجموعه ای از سیستم‌های متصل به هم را مدیریت می کند.

به علت تأخیر های انتقال در شبکه و نویز های احتمالی در خطوط انتقالی قابلیت اعتماد اجرای یک برنامه در یک سیستم تنها، بیشتر از قابلیت اعتماد اجرای آن در یک سیستم توزیع شده است. همچنین در سیستم توزیع شده اگر یکی از کامپیوتر هائی که وظیفهٔ اصلی برنامه جاری را بر عهده دارد خراب شود کل عمل سیستم مختل خواهد شد. از طرف دیگر اگر اطلاعاتی همزمان در چند کامپیوتر به صورت یکسان ذخیره گردد و یکی از کامپیوتر ها خراب شود، داده ها را می توان از کامپیوتر های دیگر بازیابی کرد و از این نظر امنیت افزایش می یابد؛ تذکر: به سیستم های توزیع شده گاهی اوقات سیستم های Loosely Coupled با ارتباط ضعیف نیز می گویند، چرا که هر پردازنده کلاک و حافظه مستقلی دارد. پردازنده ها از طریق خطوط مخابراتی مختلفی مثل گذرگاه‌های سریع یا خطوط تلفن با همدیگر ارتباط دارند.

چند دلیل برای استفاده از سیستم های گسترده عبارتند از:

1- اشتراک منابع

2- تسریع محاسبات

3- قابلیت اعتماد: اگر در سیستم توزیع شده، کامپیوتری خراب شود، کامپیوترهای دیگر (در صورت طراحی سیستم برای این منظور) می توانند کار را ادامه دهند. ولی همانطور که قبلاً گفتیم اگر سیستم نتواند خطا را تشخیص داده و آن را به کامپیوتر دیگری بسپارد بر عکس امنیت کاهش می یابد.

4- ارتباطات (مثل پست الکترونیکی و انتقال فایلها)

نخ (thread) و سیستم های multitasking

در تکنیک چند نخی (Multitasking) یک فرایند (process) که برنامه ای در حال اجراست، می تواند به بخشها و یا نخهائی (بندهائی) تقسیم شود که می توانند به صورت همزمان اجرا شوند.

برنامه هایی که چند وظیفه مستقل از هم را انجام می دهند می توانند به صورت چند نخی نوشته شوند. گاهی اوقات به سیستم های multitasking سیستم های چند تکلیفی یا چند وظیفه ای (multitasking) هم گفته می شود.

فرآیند (process) یا پردازش اساساً یک برنامه در حال اجراست که منابعی از سیستم به آن تخصیص داده شده است. (شامل رجیستر ها، حافظه، فایلها و دستگاهها) فرآیند می تواند مجموعه ای از یک یا چند نخ باشد. به نخ، رشته یا بند هم گفته می شود. کلیه اطلاعات مربوط به هر پروسس، در یکی از جداول سیستم عامل به نام جدول Process

Control Block – PCB ذخیره می شود. این جدول یک آرایه یا لیست پیوندی از ساختار هاست که هر عضو آن مربوط به یکی از پروسس هاست که در حال حاضر موجودیت دارد.

اطلاعات موجود در PCB عبارتند از: حالت جاری پردازش، شماره شناسایی پردازش، اولویت پردازش، نشانی حافظه پردازش، نشانی محل برنامه پردازش بر روی دیسک، نشانی سایر منابع پردازش، محلی برای حفظ ثباتها، در فصل سوم PCB را کامل تر شرح می دهیم.

سیستم های چند پردازنده ای (multiprocessor)

کامپیوتر ها می توانند به جای یک CPU چندین CPU داشته باشند که در اینصورت به آنها سیستم های multiprocessor می گویند. جهت استفاده از این سیستم ها نیاز به یک سیستم عامل خاص می باشد که بتواند چندین برنامه (یا نخهای یک فرآیند) را به صورت موازی واقعی روی آنها اجرا کند. سیستم عامل multitasking برای اجرا چند نخ بر روی یک CPU و سیستم عامل multiprocessor برای اجرای چند نخ بر روی چند CPU به کار می روند. گاهی اوقات به سیستم multiprocessor، سیستم multiprocessing هم می گویند.

در سیستم چند پردازنده ای، CPU ها باید بتوانند از حافظه، امکانات ورودی و خروجی و گذرگاه Bus سیستم به صورت اشتراکی استفاده کنند. مزایای این سیستم ها عبارتند از:

الف) زیاد شدن توان عملیاتی (throughput)، منظور از throughput تعداد کارهائی است که در یک واحد زمانی تمام می شوند. بدیهی است هر چقدر تعداد پردازنده ها بیشتر باشد تعداد کارهای تمام شده در یک پریود زمانی نیز بیشتر خواهد بود. البته این نسبت خطی نیست، مثلاً اگر تعداد پردازنده ها n باشد سرعت اجرا برنامه ها n برابر نمی شود چرا که بخشی از وقت پردازنده ها جهت مسائل کنترلی و امنیتی و سوئیچ کردن ها به هدر می رود.

ب) صرفه جوئی در هزینه ها، از آنجا که پردازنده ها منابع تغذیه، دیسکها، حافظه ها و ادوات جانبی را به صورت مشترک استفاده می کنند در هزینه های سخت افزاری صرفه جوئی می شود.

ج) تحمل پذیری در برابر خطا (fault – tolerant)، سیستم های مالتی پروسسور قابلیت اعتماد را افزایش می دهند چرا که خرابی یک CPU سبب توقف سیستم نمی شود بلکه تنها سبب کند شدن آن خواهد شد استمرار عمل با وجود خرابی نیازمند مکانیزمی است که اجازه دهد خرابی جست و جو شده، تشخیص داده شده و در صورت امکان

اصلاح شود (یا کنار گذاشته شود). این توانایی به ادامه سرویس، متناسب با سطح بقای سخت افزار، تنزل مطبوع یا graceful degradation نامیده می شود.

سیستم عاملهای چند پردازنده ای به دو دسته کلی متقارن و نامتقارن تقسیم می شوند.

در سیستم چند پردازنده ای نامتقارن (ASymmetric Multi Processing = ASMP) یک پردازنده جهت اجرا سیستم عامل و پردازنده های دیگر جهت اجرای برنامه های کاربران استفاده می شود. از آنجا که کد سیستم عامل تنها روی یک پروسسور اجرا می شود، ساخت این نوع سیستم عامل نسبتاً ساده است و از تعمیم سیستم عامل تک پردازنده ای به دست می آید این نوع سیستم عامل ها برای اجرا روی سخت افزار های نامتقارن مناسب هستند. مانند کمک پردازنده و پردازنده ای که به هم متصل هستند یا دو پردازنده ای که از تمام حافظهٔ موجود مشترکاً استفاده نمی کنند. یکی از معایب سیستم عامل نامتقارن غیر قابل حمل بودن (non – portable) آن است. یعنی برای سخت افزار های مختلف باید سیستم عاملهای مختلفی نوشته شود چرا که نامتقارنی می تواند حالات مختلف داشته باشد.

در سیستم چند پردازنده ای متقارن (Symmetric Multi-Processing =SMP) سیستم عامل می تواند روی هر یک از پروسسور های آزاد یا روی تمام پردازنده ها همزمان اجرا شود. در این حال حافظه بین تمام آنها مشترک می باشد. تمام پردازنده ها اعمال یکسانی را می توانند انجام دهند. سیستم متقارن از چند جنبه نسبت به نوع نامتقارن برتری دارد.

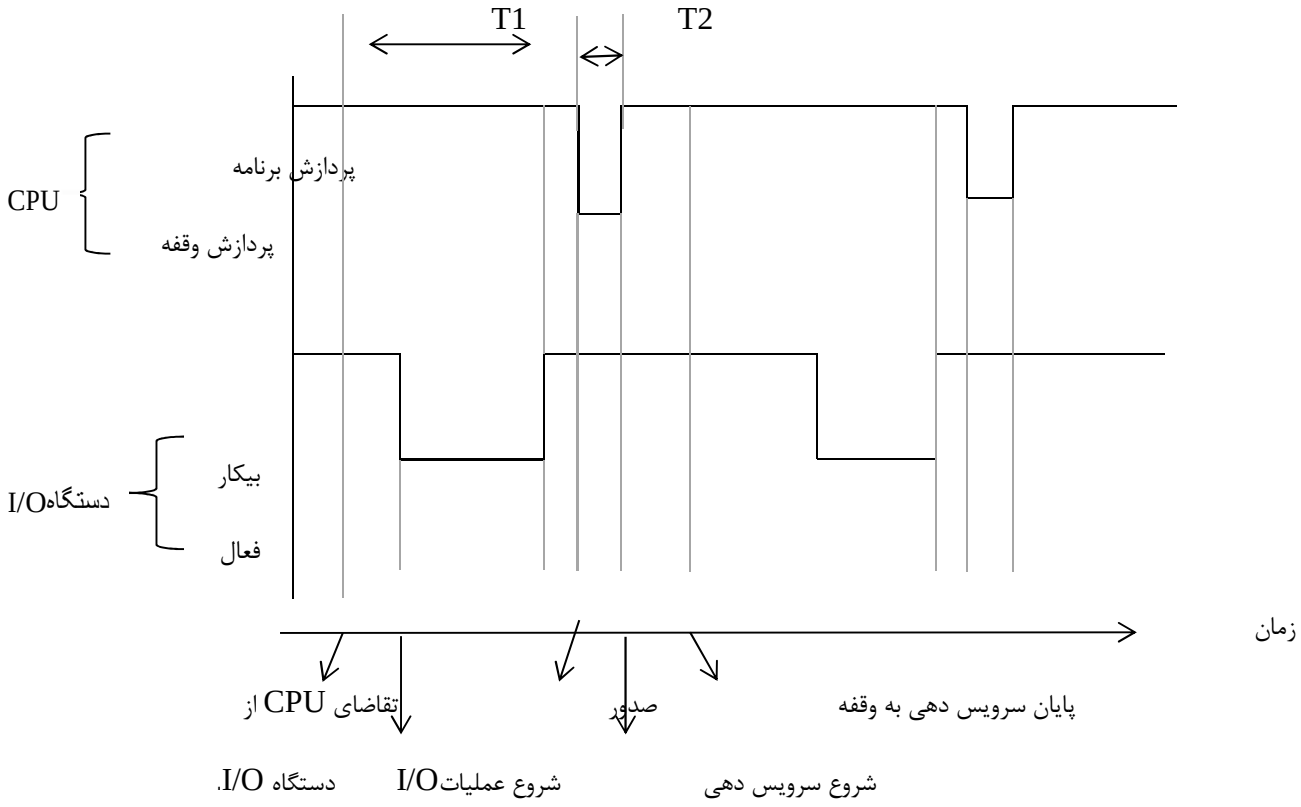
1- CPU دستگاه جانبی را مقدار دهی می کنند(مثلاً تعداد با یتهای که باید خوانده شود یا آدرس بافر)و سپس تقاضای شروع عمل I/O را به دستگاه می فرستد.

2- پردازنده به سراغ پردازش دیگری می رود و دستگاه جانبی نیز همزمان کار خود را آغاز می کند.

3- دستگاه I/O پس از اتمام کار با فرستادن سیگنال وقفه این موضوع را بهCPU اعلام می کند.

4-پردازنده در اولین فرصت به وقفه صادر شده سرویس می دهد.یعنی مثلا در مورد دستگاه کی بورد و پس از بروزوقفه، داده ورودی را از بافر کی بورد خوانده و کار پردازش را ادامه می دهد.

شکل زیر این مراحل را نشان می دهد:



وقفه توسط CPU.

در دستگاهای جانبی کند زمان پردازش وقفه(T2) خیلی کوجکتر از زمان آزاد بودن CPU برای سایر پردازش ها(T2) است.بنابر این زمان پردازش وقفه در کارئی پردازنده اثری نمی گذارد.ولی در دستگاهای سریع ممکن است T2 قابل مقایسه باT1 باشد و این امر بر عملکرد پرازنده اثر میگذارد.برای رفع این مشکل غالباً در دستگاهای جانبی سیع از تکنیکDMA (Direct Memory Access)استفادهمیشود.در تکنیک DMA دستگاها I/O پس از تنظیم ، بلوک بزرگی از داده ها را در زمان طولانی ترT1(نسبت به T2) به یکباره منتقل کرده وسپس وقفه ای را به CPU می فرستد این انتقال توسط یک کنترلر سخت افزاری

DMA ، کنترل شده ونیازی به کنترل CPU ندارد.بدین ترتیب می توان بلوک بزرگی از داده را بدون احتیاج به نظارت دائم منتقل ساخت.

تذکر1: برای اینکه چند برنامگی کارهای Io-Limited وCPU-Limited مؤثر باشد وجود مکانیزم وقفه لازم وضروری است.در این حال برنامه I/O-Limited دستگاه جانبی را کنترل می کند . وقتی که دستگاه جانبی در حال کار است برنامه متوقف شده و وقوع وقفه باعث می گردد با اتمام انتقال، کنترل دوباره از کارCPU -Limited به کار I/O-Limited برگردد.

تذکر2: مکانیزم سرویس دهی وقفه ها در کامپیوتر های مختلف، متفاوت است.

تذکر 3:روتین های مربوط به وقفه ها باید سریع باشند لذا معمولاً به زبان ماشین وبه صورت بب می شوند.

تذکر4: اگر چند منبع در یک لحظه همزمان سیگنال وقفه را ارسال کنند،CPU با تکیکهائی آن ها را اولویت بندی کرده وسپس بر اساس اولویت سرویس می دهد.

PSW

قبل از آنکه سیستم عامل کنترل را به یک روال وقفه گیر بفرستد، وضعیت پردازش جاری را در محلی حفظ می کند تا بعداً بتواند آن را دامه دهد. به این عملیات Context switch یا تعوض متن گفته می شود.برای عملیات تعویض متن از ثباتهای داخلی PSW(Program Status Word) که ترتیب اجرای دستورات را کنترل کرده و حاوی اطلاعات مختلفی می باشند، استفاده می گردد.

فراخوان های سیستمی (System Calls)

از یک جنبه دستورات در سیستم عامل را می توان به دو دسته تقسیم بندی کرد:الف)دستورات مربوط به پوسته سیستم عامل مثل dir یا del در سیستم عامل 0Dos ب)دستورات فراخوانی سیستم که اغلب درون برنامه استفاده می شوند.

فرخوان های سیستمی رابط ما بین سیستم عامل و برنامه های کاربردی می باشند . در زبانسطح بالای C و پاسکال مستقیماً می توان این فراخوان های سیستمی را به کار برد.تعدادی از خوان های سیستمی عبارتن از:

1-مدیریت پردازشها:مانند ایجاد واتمام پردازش ، بارگذاری و اجرای پردازش، تخصیص وآزاد کردن حافظه و غیره.(..,exit,wait,fork)

2-مدیریت فایلها و فهرستها: ایجاد و حذف فایل، باز و بسته کردن فایل ، خواندن و نوشتن ، تغیر صفات فایل و غیره.(...,close,write,read,open,creat).

3-مدیریت وسایل:درخواست و رهاسازی وسیله ، خواندن و نوشتن در وسیله و غیره.

4-بدست آوردن اطلاعات:خواندن و تنظیم تاریخ و زمان ، خواندن زمان استفاده از سیستم توسط کاربر، تعداد کاربران،میزان فضای آزاد حافظه یا دیسک، نسخه سیستم عامل و غیره.

اکثر سیستم عاملها (مثل DOS وUNIX) به وسایل I/O مشابه فایلها نگاه می کنند و ابزر های I/O با نامهای فایلهای ویژه شناخته می شوند.دراین حال برای کار با وسایل I/O می توان از همان دستورات read و write فایلها استفاده کرد.

تذکر: سیستم عامل Minix ،53 فراخوان سیستمی دارد که به 6 رشته دسته تقسیم می شوند:

1-مدیریت پردازش ها2-مدیریت فایل ها3-مدیریت فهرست ها 4-مدیریت زمان 5- مدیریت سیگنالها 6- مدیریت حفاظت.

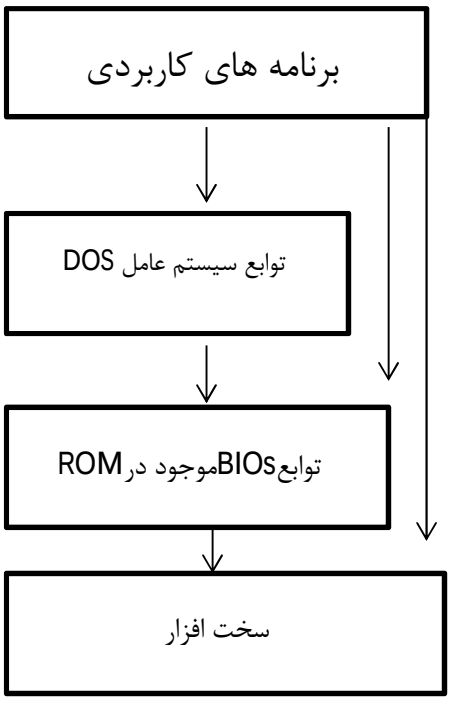
انواع سیستم عامل از نظر ساختاری

چهار ساتاری که برای ایجاد سیستم عاملها که در عل آزمایش شده اند عبارتند از :1- سیستم های یکپارچه (Monolithic) 2- سیستم لایه ای (Layered system) 3- ماشین های مجازی (Virtual Machine) 4- سیستم های مشتری-خدمت گزار(Client-Server).

1-سیستم های یکپارچه

سیستم های تجاری زیادی وجو دارند که ساخار خوش تعریفی ندارند.اغلب این سیستم عاملها به عنوان سیستم های کوچک وحدودی شروع شده اند و سپس به تدریج واری دید اولیه طراحان گسترش یافته اند.سیستم عاملDOS از این دسته می باشد.سیستم عامل به صورت یک مجموعه از رویه ها نوشته شده است که هریک از آنها می توانند دیگری را به هنگام نیاز فراخوانی کنند.برای مخفی کردن اطلاعات امکاناتی وجود ندارد و هر رویه برای دیگر رویه ها کاملاً قابل مشاهده است.

مثلاًدر MS_DOS واسطه ها وسطوح عملیاتی به خوبی مجزا نشده اند و طبق شکل زیر برنامه های کاربردی می توانند مستقیماً به توابع ROM BIOS ویا حتی پورت دستگاهای مختلف(مثل هارد دیسک) دسترسی پیا کنند، لذا به راحتی می توان برنامه های مخرب زیاد تحت DOS پدید آورد.

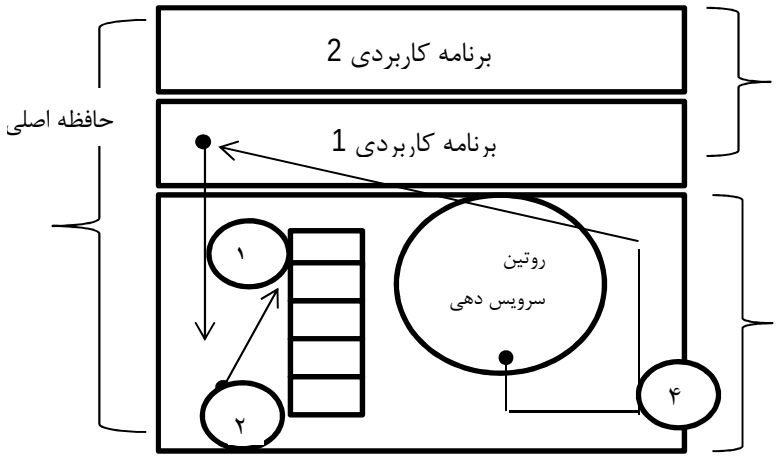


اکثر CPU ها دارای دو مد کاری هستند یکی مود هسته که مخصوص سیستم عامل است و در آن تمام دستور عمل ها

مجاز میباشد و دیگری مود کاربر است که مخصوص برنامه های کاربران بوده و در آن دستورات I/O ودستور عمل های معین دیگری مجاز نمی باشند.

سیستم عاملDOS توسط سخت افزار زمان خود(پردازنده8088) محدود بوده است چرا که این پردازنده فقط در یک مدکار می کند و دستورات در آن مجاز میباشد ولی پردازنده 386 دارای مد های مختلفی است که سیستم عامل ویندوز از آن به خوبی استفاده می کند.

ساختار کلی این سیستم عاملها مشابه شکل زیر است:



برنامه ای کاربردی یکی از فراخوانهای سیستـمی(توابع سیستم عامل)را صـدا می زند. در این حـال ماشین از مد کاربر (user mode) به مد هسته (Kernel mode) تغیرحالت می دهد وکنترل به سیستم عامل سپرده میشود(مرحله 1). سیستم عامل باتوجه به پارامتر های تابع مذکور تعیین میکند کدام فراخوان سیستمی باید اجرا شود(مرحله 2) سپس سیستم عامل به جدولی رجوع میکند که در ردیف K ام آن جدول یک اشاره گر به رویه اجراکننده فراخوان سیستمی وجود دارد.(مرحله3). سپس آن روتین اجرا شده و در انتها کنترل به برنامه کاربر بر میگردد(مرحله4).

2-سیستم های لایه ای

در روش لایه ای سیستم عامل به تعدادی سطح یا لایه تقسیم می شود که هرکدام در بالای لایه پایین تر قرار میگرند.مزیت مهم این روش پیمانه ای (modularity) بئدن آن است .یعنی لایه ها به گونه ای تقسیم بندی میشوند که هر

لایه فقط توابع و سرویس های لایه پایین تر را استفاده می کند.

بدین ترتیب هر لایه را می توان مستقل از لایه های دیگر طراحی کرد،بسط داد و خطایابی کرد.

هر سطح با استفاده از اعمال لایه های پایین تر پیداه سازی میشود ولی آن سطح نمی داند که اعمال سطح پایین چگونه پیاده شده اند و فقط باید بداند که آن اعمال چه می کنند .بدین ترتیب هر لایه مسائلی را از لایه های بالاتر مخفی میسازد.

اولین سیستم لایه ای ، سیستم THE با 6 لایه بود: لایه صفر مسائل زمانبندی (scheduling)پردازنده را انجام می دهد یعنی اینکه در هر لحظه CPU در اختیار کدام برنامه باشد.لایه یک مدیریت حافظه(اصلی و جانبی) را بر عهده دارد. لایه دو ارتباط بین هر پروسس و کنسول اپراتور را برقرار میسازد.

لایه	وظیفه
5	The operator
4	User programs
3	I/O management
2	Operator-Process Communication
1	Memory management
0	CPU scheduling

لایه سه مدیریت دستگاههای I/O وبافر کردن اطلاعات را برعهده دارد. در بالای این لایه هر پروسس به جای دستگاههای I/O حقیقی و پیچیده با دستگاههای ساده و مجازی I/O سرو کار دارد.در لایه چهار برنامه های کاربران اجراء می شوند که هیچ نگرانی در مورد مدیریت در مورد مدیریت پروسس ، حافظه، کنسول و I/O ندارند.در لایه پنجم پروسس اپراتور سیستم قرار می گیرد.

مشکل اصلی در روش لایه لایه، تعریف مناسب لایه های مختلف است.از آنجا که یک لایه فقط می تواند لایه های پائین تر را به کار بر برای طراحی آن باید دقت زیادی به خرج داد.مشکل دیگر این ساختار این است که نسبت به انواع دیگر بازدهی کمتر دارند.هنگامی که دستورات از لایه بالا به سمت پایئن حرکت می کنند، در هر پارامتر های دستور ممکن است، از نظر صحت برسی شده ویا تغیر یابند .لذا هر لایه قدری باسر(overhead) به سیستم اضافه میکند و در نتیجه فراخوانی سیستمی نسبت به سیستم غیر لایه ای بیشتر طول میکشد.لذا در سالهای اخیر سعی شده است لایه های کمتری با قابلیت عمل بیشتری طراحی شود . به عنوان مثال محصول اولیه Windows NT با لایه های زیاد ،

کارایی کمتری نسبت به ویندوز95داشت.در NT 4.0 سعی شده لایه ها به همدیگر نزدیکتر و مجتمع تر شود تا کارایی بیشتر گردد.

سیستم MULTICS به جای لایه ها به صورت یکسری لقه های متحدالمرکز سازماندهی شده است.بطوریکه هر حلقه داخلی از امتیازات بالاتری نست به حلقه خارجی خود بهرمند می باشد.اگر یک رویه از حلقه خارجی بخواهد یک رویه از حلقه داخلی را صدا بزند، باید یکی از فراخوان های سیستمی را اجراء کند و اعتبار پارامتر های این دستورالعمل قبل از اجراء به دقت برسی می شود.مثلاً یک استاد برنامه گرفتن امتحان و نمره دادن را در حلقه n می نویسد و برنامه دانشجویانش در حلقه a+1 اجراء می شود، بدین ترتیب دانشجویان نمی توانند نمره خود را تغیر دهند.

3-ماشین مجازی

سیستم عامل VM بر روی سیستم های IBM بهترین مثال از مفهوم ماشین مجازی است. قلب سیستم که به مانیتور ماشین مجازی (Virtual Machine Monitor) معروف است ، بر روی سخت افزار عریانی اجراء شده و چند برنامگی را پدید می آورد، این مانیتور چندین ماشین مجازی را در لایه بالاتر فراهم می سازد.

برنامه کاربر3	برنامه کاربر 2	برنامه کاربر 1
CMS	CMS	CMS
مانیتور ماشین مجازی		
سخت افزار		

این ماشین های مجازی برای کاربران مشابه یک نسخه از سخت افزار عریان هستند که دارای مودهای کاربر و هسته ، I/O ، وقفه ها و چیزهای دیگر «ماشین حقیقی» می باشند. به هر کاربرماشین مجازی خودش داده میشود و او میتواند هر یک از سیستم عامل ها یا بسته های نرم افزاری موجود را روی ماشین خودش اجراء کند.

هر کاربر یک برنامه CMS (Conversational Monitor System) مخصوص به خود را دارد که یک سیستم عامل تک کاربره محاوره ای است. مزایای این ماشین مجازی عبارتند از:

الف) در این سیستم دو وظیفه اصلی چندبرنامگی و ایجاد واسطه راحت (مستقل از سخت افزار) از یکدیگر مجزا شده اند . مانیتور ماشین مجازی وظیفه چنربرنامگی را بر عهده دارد و لایه بالای آن وظیفه ایجاد واسطه کاربر با سخت افزار را بر عهده دارد. لذا هر یک از این بخشها ساده تر شده و از قابلیت انعطاف بیشتری برخوردارند.

ب) هر ماشین مجازی از سایر ماشین ها کاملا جداست. بنابراین هیچ مشکل امنیتی مجود نخواهد داشت و برنامه های کاربران تداخلی با همدیگر ندارند.

ج) از آنجا که هر ماشین مجازی کاملا مشابه سخت افزار واقعی است، هر یک از آنها می تواند هر سیستم عاملی را مستقلا اجرا کنند لذا می توان همزمان سیستم عاملهای مختلفی را روی این ماشین اجرا کرد. این امر همچنین باعث می شود مراحل تحقیق و توسعه سیستم عاملها راختر صورت بگیرد ، چرا که دیگر سازندگان سیستم عامل برای تست کردن سیستم عامل تولیدی جدید لازم نیست کل کامپیوتر را در اختیار داشته باشد در حالیکه سایر کاربران در

حال استفاده از سیستم هستند ، طراحان می توانند سیستم عامل جدید خود را روی یکی از ماشین های مجازی آزمایش کنند، و این کار به ندرت باعث از کار افتادن کامپیوتر می شود.

ایده ماشین مجازی امروزه نیز جهت رفع مشکلات عدم سازگاری گسترش زیادی یافته است. به عنوان مثال شرکتهای میکروسیستم یا شرکت DEC که کامپیوترهای غیر intel را می سازند مایلند که مشتریهایشان بتوانند برنامه های معروف تحت DOS (تحت intel) را نیز اجراء کنند. برای این کار یک ماشین مجازی اینتل بر روی پردازنده خود پدید می آوردند. در این حال ماشین مجازی دستورات اینتل را به دستورات پردازنده جدید تبدیل می کند. یا مثلا کامپیوتر power pc شامل ماشین مجازی Motorola 68000 می باشد. مثال دیگر اجراء شدن DOS تحت محیط ویندوز است، پردازنده های 386 به بعد دارای یک مد مجازی هستند که میتوانند چندین برنامه DOC را با هم اجراء کنند، ویندوز از این مد مجازی استفاده کرده و اجازه می دهد برنامه های تحت DOS، تحت ویندوز نیز اجراء شوند (البته به شرطی که دستورالعملهای عادی را اجراء کنند و مستقیما با پورتهای مهم سر و کار نداشته باشند).

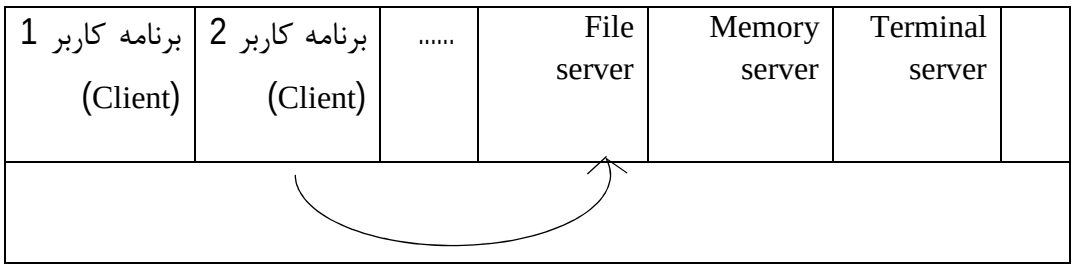
مثال دیگر از این مفهوم ماشین مجازی زبان جاوا (JAVA) می باشد. کامپایلر زبان جاوا که توسط شرکت Sun طراحی شده است یک خروجی بایت کد (Byte code) تولید میکند. این بایت کد ها دستوراتی هستند که بر روی ماشین مجازی جاوا(JVM) اجراء می شوند. جهت اجرای برنامه های جاوا در یک ماشین ، آن کامپیوتر میبایست دارای یک JVM باشد . امروزه JVM ر روی بسیاری از انواع کامپیوتر ها (PC، مکینتاش، Sun، مینی کاپیوترها و مین فریم ها) وجود دارد. JVM همچنین در Microsoft explorer ویندوز پیاده سازی شده است. بدین ترتیب برنامه هایی که به زبان JAVA نوشته شده اند به راحتی بر روی انواع کامپیوتر اجراء میشوند. فقط کافی است بایت کد ها را روی آن ماشین کامپایل کرد بدیهی است به علت نیاز به کامپایل شدن بایت کدها، برنامه های جاوا سرعت کمتری نسبت به برنامه هایی نظیر C دارد. برنامه های C توسط کامپایلر بومی یک کامپیوتر ، برای یک بار تبدیل به زبان ماشین آن کامپیوتر می گردد. پس خروجی زبان ماشین کامپایلر C از یک نوع کامپیوتر به کامپیوتر دیگر متفاوت است ولی بایت کد های خروجی جاوا برای همه ماشین ها یکسان است.

4- سیستم مشتری – خدمتگزار (client - server)

سیستم عامل VM با جا به جا کردن بخش زیادی از کد سیستم عامل به لایه بالاتر (یعنی CMS) باعث ساده شدن هسته اصلی یعنی مانیتور ماشین مجازی شد. با این همه هنوز هم VM یک برنامه پیچیده می باشد.

روند طراحی سیستم عاملهای جدید همواره با این ایده همراه بوده که تا جایی که ممکن است کد ها به لایه های بالاتر منتقل شوند تا نهایتا یک هسته کمینه پدید آید. برای این منظور اکثر وظایف سیستم عامل را در سطح کاربر و مشابه پردازشهای کاربران پیاده سازی می کنند. مثلا برای خواندن یک بلوک از فایل پروسس کاربر (پروسس مشتری client) یک درخواست به پروسس خدمتگزار (server) ارسال می کند و از آن می خواهد که کارش را انجام داده و

جواب را برگرداند. در این مدل تنها کاری که هسته انجام میدهد این است که ارتباط بین مشتری و خدمتگزار را از طریق پیام ها برقرار می سازد (مشابه شکل زیر):

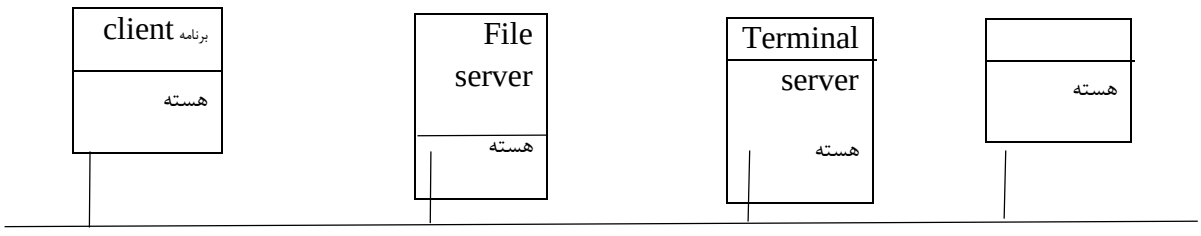


مزایای این مدل عبارتند از :

الف) از آنجا که سیستم عامل به چند بخش تقسیم شده که هر یک فقط یکی از وظایف سیستم عامل را انجام می دهند ، بنابراین سیستم عامل را می توان ساده تر طراحی و پیاده سازی کرد.

ب) به علت اجرای کلیه پروسس های خدمتگذار در مد کاربر (و نه در مد هسته) هیچکدام از آنها دسترسی مستقیم به سخت افزار ندارند ، لذا اگر اشکالی مثلا در خدمتگزار فایل ایجاد شود فقط موجب اختلال در خدمت فایل خواهد شد و به ندرت مجب خراب شدن کل سیستم می شود.

ج) فایده دیگر این مدل سازگاری آن برای استفاده در سیستم های توزیع شده است. از آنجا که یک مشتری به وسیله ارسال پیام هایش با یک خدمتگزار ارتباط برقرار می کند، مشتری نیاز ندارد که بداند آیا به پیغام وی به صورت محلی در ماشین خودش رسیدگی می شود و یا اینکه پیغام از طریق یک شبکه به یک ماشین دور ارسال می شود(مشابه شکل زیر):



(مدل Client – server در یک سیستم توزیع شده)

مکانیزم و سیاست

یک اصل مهم در طراحی سیستم عاملها جداسازی سیاست (policy) از مکانیزم (mechanism) می باشد. مکانیزم چگونگی نحوه انجام کاری را نشان می دهد ولی سیاست آنچه را که باید انجام شود نشان می دهد. مثلا اینکه «چگونه» CPU به کاربری داده شود یک مکانیزم است ولی اینکه «چه» مدت زمانی باید CPU داده شود یک سیاست است.

سیاستها احتمالا از ماشینی به ماشینی دیگر و از زمانی دیگر تغییر می کنند. در بدترین حالت هر تغییردر سیاست سبب یک متغیر در مکانیزم مربوطه اش می شود. ولی در بهترین حالت یک تغییر در سیاست تنها نیازمند تعریف مجدد پارامترها می گردد.

یکی از ایرادات این سیستم آن است که تفاوتی بین CMS ها از نظر حجم کاری که انجام می دهند قائل نشده و به همه به یک نسبت مساوی سرویس داده می شود. حال آن که ممکن است یک CMS کار سبک و یک CMS دیگر کار سنگینی داشته باشد. برای رفع این مساله در برخی نسخه های VM بعضی عملیات سنگین I/O توسط نرم افزار های ویژه انجام گرفته و همچنین می توان CMS ممتاز تعریف کرد.

سیستم عامل های بر ریز هسته و نیز مبتنی بر client-server جدایی مکانیزم و سیاست را حتی الامکان در نظر میگیرند.

زبان پیاده سازی سیستم عاملها

سیستم عاملهای اولیه به زبان اسمبلی نوشته می شدند ولی امروزه اکثر سیستم عاملها به زبان c (یا c++) نوشته میشوند. سیستم عامل UNIX، OS/2 و ویندوز بیشتر به زبان C نوشته شده اند و قسمت اندکی از آنها به زبان اسمبلی است . مهمترین مزیت استفاده از زبان سطح بالا برای پیاده سازی سیستم عامل قابلیت حمل آن بر روی انواع کامپیوتر ها و سادگی پیاده سازی ، تغییر و بسط دادن سیستم عامل میباشد.

ممکن اسن ادعا شود پیاده سازی سیستم عامل به زبان c باعث کاهش سرعت وافزایش مصرف حافظه می گردد. اگرچه یک برنامه نویس ماهر زبان اسمبلی، میتواند برنامه های کوچک و بسیاربهینه بنویسد ولی برای برنامه های بزرگ یک کامپایلر خوب، می تواند تحلیل پیچیده تری نسبت به مغز انسان ماهر انجام داده و بیهنه سازی های کاملی را انجام دهد. لذا در عمل برنامه های بزرگ C کد اسمبلی بینه تر و کمتری را تولید می کنند ، نسبت به حالتی که برنامه نویس بخواهد همان کارها به زبان اسمبلی انجام دهد . از طرف دیگر در عمل کارایی اصلی نتیجه

ساختمان داده و الگوریتم های بهتر استنه نتیجه نوشتن برنامه به زبان اسمبلی، همچنین اگر چه سیستم عاملها برنامه های بزرگی هستند ولی تنها بخش کوچکی از کد آنها نسبت به کارایی، بحرانی (critical) میباشد مثل مدیریت حافظه و زمانبندی CPU ، لذا پس از آنکه سیستم عامل به زبان سطح بالا نوشته شد و به درستی عمل کرد می توان روتین های گلوگاه (bottleneck) و مهم را شناسائی کرد و سپس آنها را با روتین های معادل زبان اسمبلی جایگذین نمود.

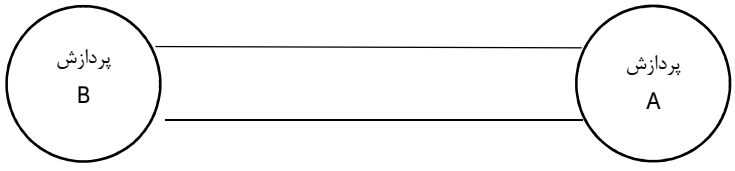
(Job Control Language) JCL

در سیستم های دسته ای (Batch) قدیمی ، کاربر نیازهای برنامه خود را از طریق یکسری دستور به نام JCL به سیستم می داد. سیستم عامل نیز بر مبنای دستورات JCL منابع مورد نیاز را در اختیار آن برنامه می گذاشت. به عبارتی دیگر اطلاعات یک job به صورت یک بسته متشکل از JCL ، برنامه و داده ها (JCL+ Program +Data) به سیستم .داده شده و دیگر کاربر ارتباط و محاوره ای با سیستم و برنامه نداشت . می توانید تصور کنید JCL چیزی شبیه batch file ها در سیستم عامل DOS است. بنابر این JCL حاوی اطلاعاتی است که کاربر به سیستم عامل می دهد و به آن می گوید چه کار کرده و از چه منابعی استفاده کند. به عبارتی دیگر در سیستم های دسته ای JCL نوعی زبان برنامه نویسی می باشد که برنامه نویس برنامه های کاربران ، دستگاه های IO و منابع مورد نیاز آن ها را به سیستم عامل معرفی می کند.

هنگامی که کار های (JOB) مختلفی به سیستم داده میشود اسپولر (SPOOLER) کارها را دسته بندی کرده و هر job را که شامل JCL + program +Data است را در قسمتی از دیسک به نام spool Area قرار میدهد. جدولی به نام «به نام جدول ورودی اسپول» یا ISPT (Input Spool Table) وجود دارد که هر سطر آن مربوط به اطلاعات کنترلی یک job است.

خط لوله (Pipeline)

لوله ها چیزی هستند که پردازش ها یا فایل ها را به همدیگر مرتبط می کنند. لوله مشابه یک فایل مجازی جهت اتصال و رد و بدل بین دو پردازش استفاده می گردد. هنگامی که پردازش A می خواهد برای پردازش B اطلاعاتی بفرستد آن را مشابه یک فایل خروجی در لوله می نویسد. سپس پردازش B برای دریافت اطلاعات این لوله را مشابه یک فایل ورودی می خواند:



خط لوله

مثلا در سیستم عامل DOS در دستوری مثل TYPE ALL.TXT|MORE دستور TYPE خروجی خود را به دستور MORE می دهد و محتویات فایل ALI.TXT به صورت صفحه صفحه چاپ می شود.