

فصل دوم

نگاه کلی به سیستم عامل

وحید رنجبر

پائیز ۹۸

مباحث این فصل:

- اهداف و وظایف سیستم عامل
 - سیستم عامل به عنوان رابط بین کاربر و کامپیوتر
 - سیستم عامل به عنوان مدیر منابع
 - سهولت تکامل تدریجی سیستم عامل ها
- تکامل تدریجی سیستم عامل ها
 - پردازش ردیفی
 - سیستم عامل های دسته ای ساده
 - سیستمهای چند برنامه ای دسته ای
 - سیستم های اشتراک زمانی
- دستاوردهای اصلی
 - فرایندها
 - مدیریت حافظه
 - حفاظت و ایمنی اطلاعات
 - زمانبندی و مدیریت منابع
 - ساختار سیستم

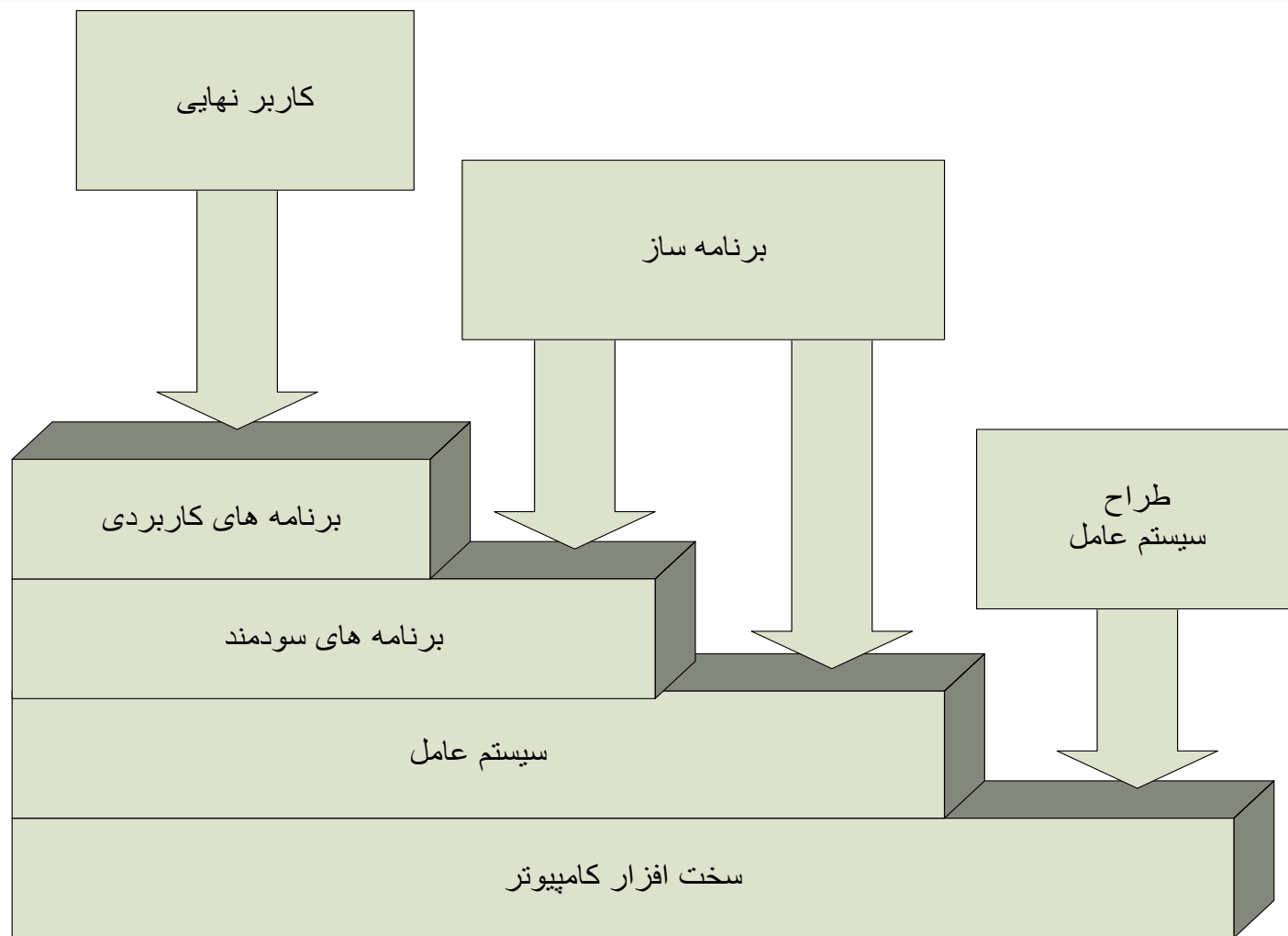
سیستم عامل چیست؟

- سیستم عامل یک برنامه است که اجرای برنامه های کاربردی را کنترل میکند .
- سیستم عامل بصورت یک رابط میان سخت افزار و کاربر عمل میکند.
- ۳ هدف سیستم عامل:
 - سهولت: OS استفاده از کامپیوتر را آسان میکند.
 - کارامدی: OS موجب استفاده کارآمد تر از منابع سیستم میشود.
 - قابلیت رشد: OS باید به گونه ای باشد که قابلیت رشد داشته باشد.

سیستم عامل : رابط بین کاربر و کامپیوتر

- کاربر نهایی با سخت افزار سروکار ندارد.
- یک کاربرد توسط برنامه ساز ایجاد میشود.
- برنامه های سودمند برنامه هایی هستند که به دفعات استفاده میشوند. و به ایجاد برنامه، مدیریت پرونده ها و کنترل I/O کمک می کنند.
- **مهمترین برنامه سیستمی سیستم عامل است،** که به صورت میانجی بین سخت افزار و برنامه ساز است.

لایه ها و منظرهای یک سیستم کامپیوتری:



برنامه های سودمند

- مجموعه برنامه های سیستمی هستند.

- توابعی هستند که به دفعات استفاده می شوند و به ایجاد برنامه مدیریت پروژه و کنترل دستگاه های ورودی و خروجی کمک می کنند.

- مانند: ادیتورها، اشکالزداها و ...

خدمات سیستم عامل:

- **توسعه برنامه:** برنامه های سودمندی مثل ویراستار، اشکال زدا که از طریق OS قابل دسترسند.
- **اجرای برنامه:** OS تمام مراحل اجرا (بار کردن داده ها و دستورات در حافظه، مقدار گذاری I/O) را زمان بندی میکند.
- **دسترسی به I/O:** فراهم کردن واسطی یکنواخت برای تمام دستگاه های I/O
- **کنترل دسترسی به پرونده ها:** راهکارهای حفاظتی برای دسترسی به سیستم پرونده ها
- **دسترسی به سیستم:** در سیستم اشتراکی OS دسترسی به منابع را کنترل میکند.
- **کشف و پاسخ خطا:** عکس العمل مناسب در برابر خطاهای حین اجرا
- **حسابداری:** جمع آوری آمار استفاده از منابع

سیستم عامل به عنوان مدیر منابع

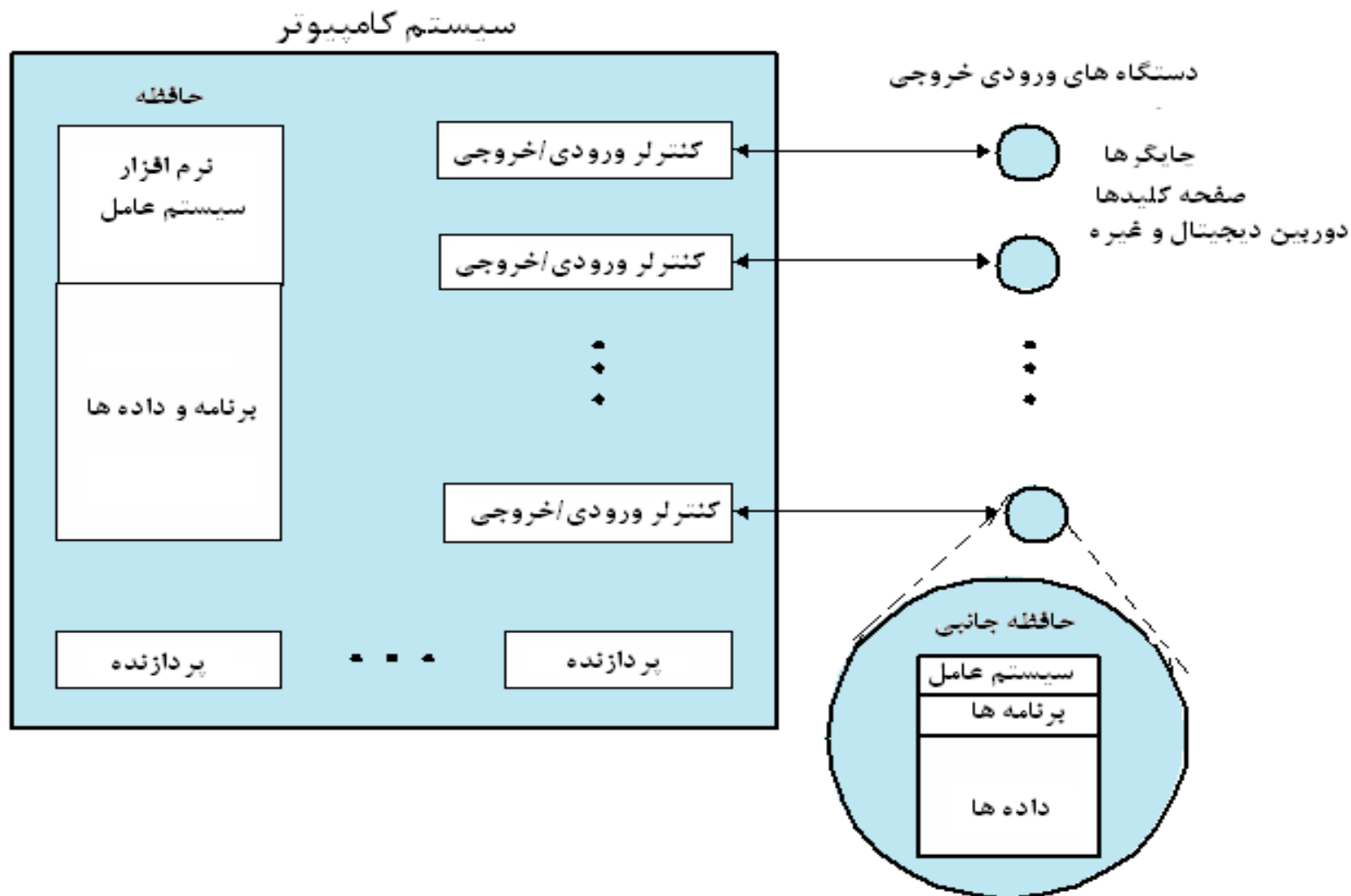
کامپیوتر مجموعه ای از منابع برای انتقال، ذخیره سازی، و پردازش داده هاست. سیستم عامل مسئول مدیریت انتقال و ذخیره سازی و پردازش داده ها را دارد.

سیستم عامل به عنوان راهکار کنترلی ازدو جهت غیر عادی است:

۱- سیستم عامل مانند نرم افزار عادی کار می کند.

۲- سیستم عامل مرتبا کنترل کردن را رها می کند.

سیستم عامل به عنوان مدیر منابع



هسته سیستم عامل :

- ✓ بخشی که در حافظه اصلی است.
- ✓ بیشترین دفعات استفاده را دارد.
- ✓ تخصیص حافظه مشترک توسط سیستم عامل و سخت افزار مدیریت حافظه در پردازنده کنترل می شود.
- ✓ در مورد زمان استفاده از I/O تصمیم می گیرد.
- ✓ میزان تخصیص پردازنده به یک کار را مشخص می کند.

دلایل تغییر سیستم عامل

۱- ارتقاء و انواع جدید سخت افزار .

این امکانات نیازمند حمایت پیچیده تری از طرف سیستم عامل است.

۲- خدمات جدید: برای پاسخگویی به نیازهای کاربران.

۳- رفع خطاهای کشف شده در سیستم عامل

تکامل تدریجی سیستم عامل

۱-پردازش ردیفی.

۲-سیتم های دسته ای ساده.

۳-سیستم های چند برنامه ای دسته ای.

۴-سیستم های اشتراک زمانی.

پردازش ردیفی

- به علت دسترسی به کامپیوتر به صورت ردیفی این نام را دارند.
- در سال ۱۹۴۰-۱۹۵۰ به وجود آمد.
- کاربر با سخت افزار در ارتباط بود.
- سیستم عامل وجود نداشت.
- دارای یک میز فرمان بود.
- به زبان ماشین و به وسیله دستگاه ورودی بارگذاری می شد.
- خروجی در چاپگر ظاهر می شد.

مسئله اصلی سیستم های اولیه

۱-زمانبندی :

هر کاربر باید از برگه های نوبت گیری استفاده کند (معمولاً نیم ساعته). در صورت کامل نشدن در موقع مقرر برنامه خاتمه می یافت تا بعداً دوباره از ابتدا اجرا شود.

۲-زمان نصب :

هر برنامه شامل بار کردن مترجم، کد منبع به حافظه، ذخیره سازی برنامه ترجمه شده بود. هر یک از این کارها متضمن قرار داشتن اطلاعات بر روی نوار ورودی بود. در صورت بروز خطا کاربر کار را از اول شروع میکرد.

سیستم های دسته ای:

- از برنامه ای به نام ناظر استفاده میشود.
- کاربر دسترسی مستقیم به ماشین ندارد. کاربر برنامه را بر روی کارت به متصدی میداد و متصدی کارتها را به طور ردیفی در دستگاه نوار خوان قرار میداد تا مورد استفاده ناظر قرار بگیرد.
- چون ناظر اکثر عملیات را انجام میدهد بخش اعظمی از آن در حافظه است که به آن ناظر ماندگار می گویند.
- پردازنده در یک زمان در حال پردازش ناظر مقیم است، با خوانده شدن کار کنترل به برنامه کاربر منتقل میشود و پس از خاتمه برنامه کنترل دوباره به ناظر بر میگردد.

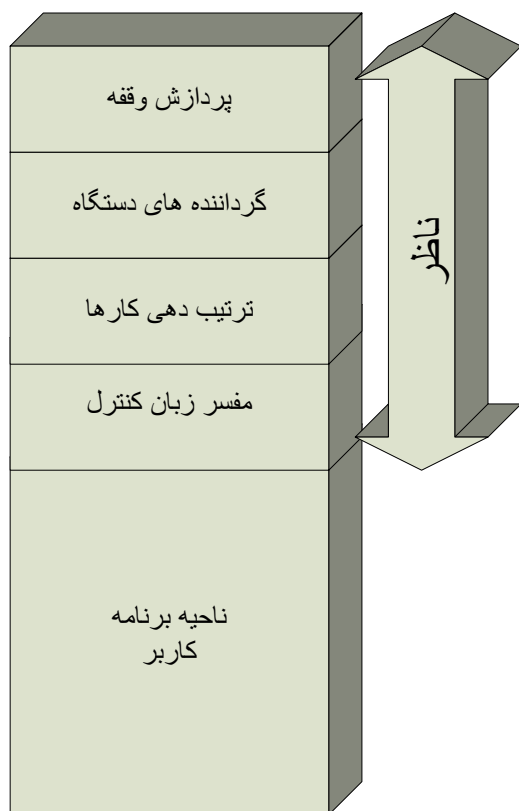
ناظر

- **قسمت اعظم ناظر:** در حافظه اصلی و آماده اجرا می باشد
(ناظر ماندگار)

- **بقیه ناظر:** برنامه سودمند و توابع عمومی و مشترک
(زیر برنامه)

زمان تنظیم شرایط اولیه کارها را ناظر انجام میدهد.

- JCL = دستورالعمل هایی از زبان کنترل کار.



وضعیت حافظه

برای یک ناظر ماندگار در 6 حافظه

ویژگی های مطلوب سخت افزاری در سیستم پردازش دسته ای:

- **حفاظت از حافظه اصلی :** ناظر ماندگار نباید در حافظه تغییر کند، در صورت چنین تلاشی پردازنده باید خطا را کشف و کنترل را به ناظر برگرداند.
- **زمان سنج:** سیستم نباید در انحصار اجرای یک برنامه باشد، با زمان سنج می توان کارها را زمانبندی کرد.
- **دستورالعمل های ممتاز:** دستورالعمل هایی که تنها توسط ناظر اجرا می شوند (مثل I/O)
- **وقفه ها:** این خصوصیت به سیستم عامل انعطاف بیشتری میدهد.

حافظت از حافظه:

- برنامه های کاربر در **حالت کاربر** اجرا می شوند.
 - بعضی دستورالعمل ها نمی توانند اجرا شوند.
- ناظر در **حالت سیستم** اجرا می شود.
 - به حالت سیستم **حالت هسته** یا **ممتاز** نیز گفته می شود.
 - دستورالعملهای ممتاز در حالت ممتاز اجرا می شوند.
 - قسمت های محافظت شده از حافظه ممکن است در این حالت در دسترس باشند

سیستم عامل چند برنامه ای دسته ای :

- با پردازش دسته ای هم به علت اینکه اکثر برنامه به اجرای دستورالعمل های I/O مربوط می شود، و عدم تطابق سرعت I/O و CPU، باز هم پردازنده اکثر وقت خود را بیکار است. اگر ناحیه کاربر چندین برنامه در حال اجرا را در خود داشته باشد میتوان در حین اجرای عمل I/O برای یک برنامه ، سیستم عامل برنامه دیگر را اجرا کند.

- **نکته :** چون سیستم عامل چند برنامه ای نیاز به مدیریت حافظه و همچنین الگوریتم های زمانبندی دارد از سیستم عامل تک برنامه ای پیچیده تر است .

سیستم های اشتراک زمانی

- در بعضی مواقع مثل پردازش تراکنش حالت محاوره ای ضروری است. لذا سیستم های اشتراک زمانی بوجود آمدند.
- چند برنامه‌ی امکان رسیدگی به کارهای محاوره ای را میدهد. وقت پردازنده بین کارها تقسیم میشود.
- پردازنده بین کاربران به اشتراک گذاشته می شود.
- داشتن کاربران متعددی که از طریق پایانه خود به طور هم زمان از سیستم عامل استفاده می کنند.
- اگر N کاربر داشته باشد هر کاربر $1/N$ از زمان مفید پردازنده استفاده می کند.

چند برنامگی دسته ای در مقابل اشتراک زمانی

اشتراک زمانی	چند برنام های دسته ای	
حداقل زمان پاسخ	حداکثر استفاده از پردازنده	هدف اصلی
فرمان هایی که از پایانه وارد می شود	دستورالعملهای زبان کنترل کار که همراه کار ارائه شده است	منبع دستورات به سیستم عامل

CTSS(Compatible Time Sharing System)

- این سیستم روی ماشینی با ۳۲۰۰۰ کلمه حافظه ۳۶ بیتی اجرا می شد. ۵۰۰۰ کلمه برای ناظر ماندگار و ۲۷۰۰۰ کلمه برای برنامه کاربر و داده هایش
- یک برنامه همیشه طوری بار می شود که از آدرس ۵۰۰۰ شروع شود بنابراین کار ناظر و هم مدیریت حافظه آسان تر می شد.
- در این سیستم در هر ۲/۰ ثانیه وقفه ای صادر می شد که موجب می شد پردازنده کنترل را به کار دیگری بدهد.
- برای حفظ وضعیت کاربر قبلی ابتدا برنامه ها و وضعیتش روی دیسک ذخیره می شد.

مثالی برای درک عملکرد CTSS :

- مثال: اگر تعداد کلمات مورد نیاز هر کار بصورت زیر باشد نحوه انجام عملیات به صورت زیر است.

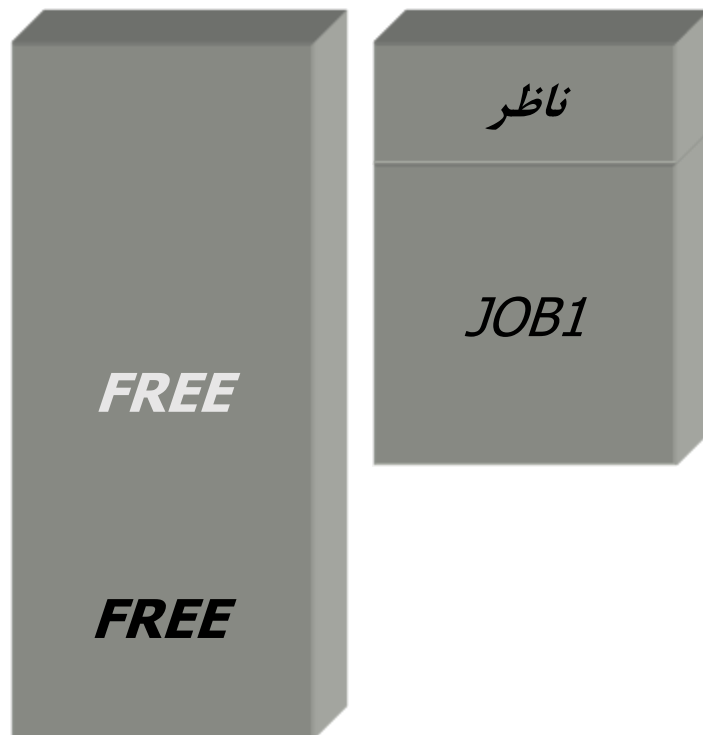
Job2: 20000 Job1: 15000 –

Job4: 10000 Job3: 5000 –

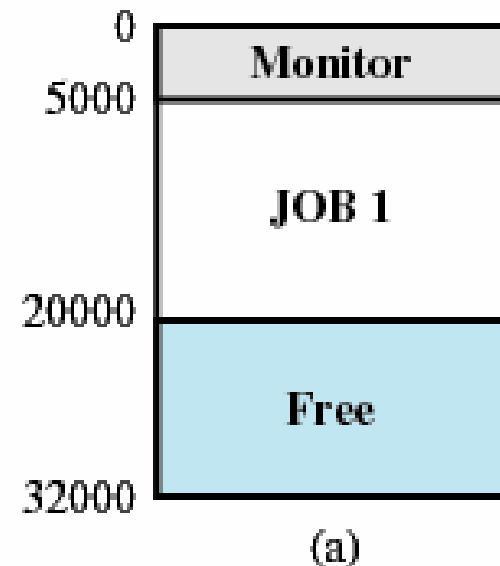
- ترتیب انجام کار ها :

Job1 \Rightarrow Job2 \Rightarrow Job3 \Rightarrow Job1 \Rightarrow Job4 \Rightarrow Job2 –

مثالی برای درک عملکرد CTSS :

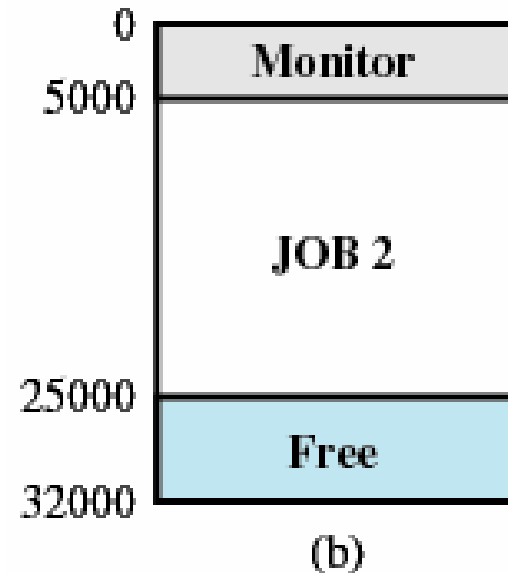
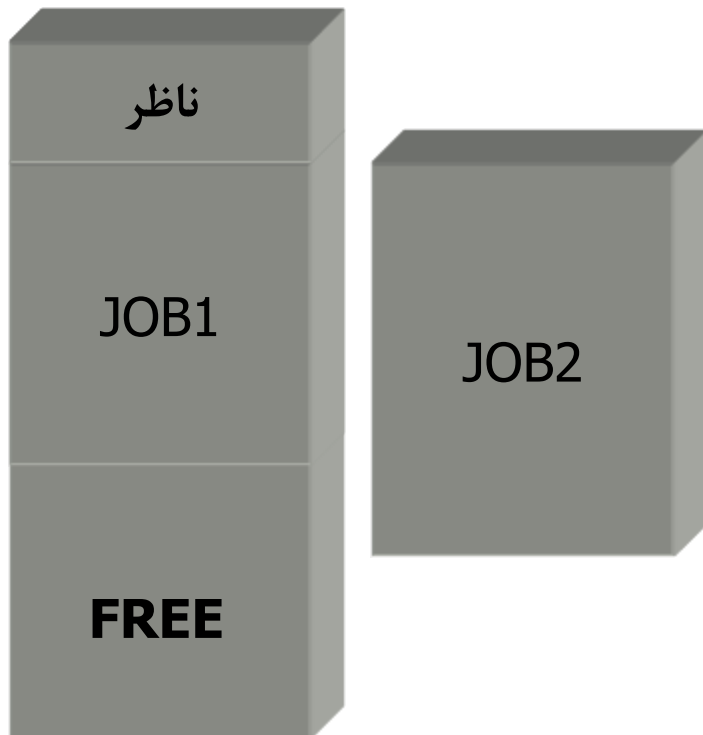


- ۵۰۰۰ کلمه اول را ناظر اشغال میکند و از محل ۵۰۰۰ Job1 بار میشود.



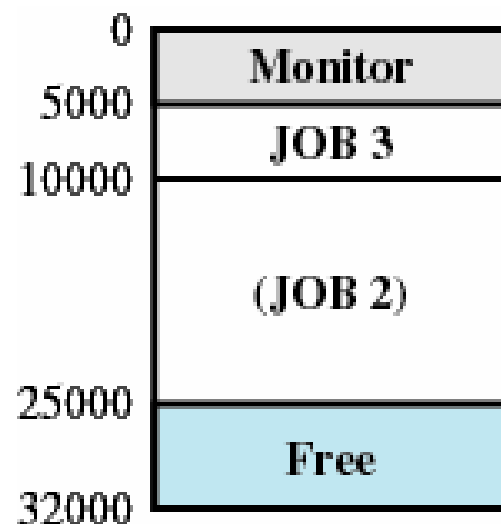
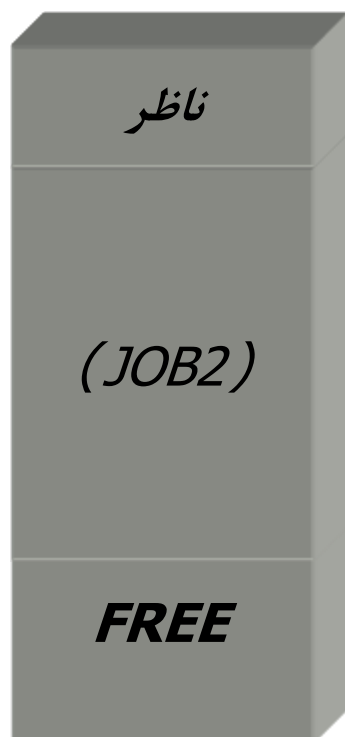
مثالی برای درک عملکرد CTSS :

- چون Job2 از Job1 بزرگتر است، کل Job1 بر روی دیسک ذخیره میشود و Job2 بار میشود.



مثالی برای درک عملکرد CTSS :

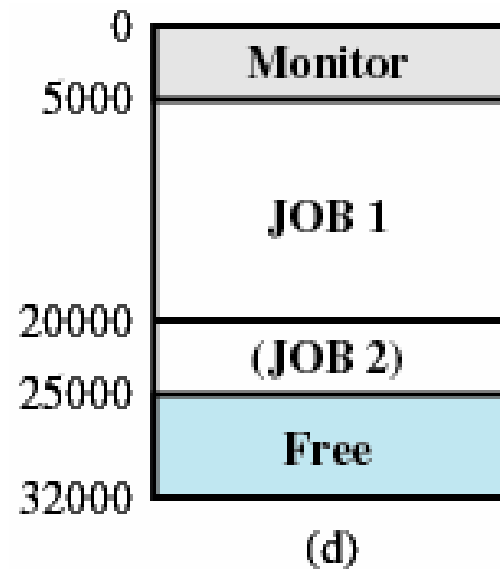
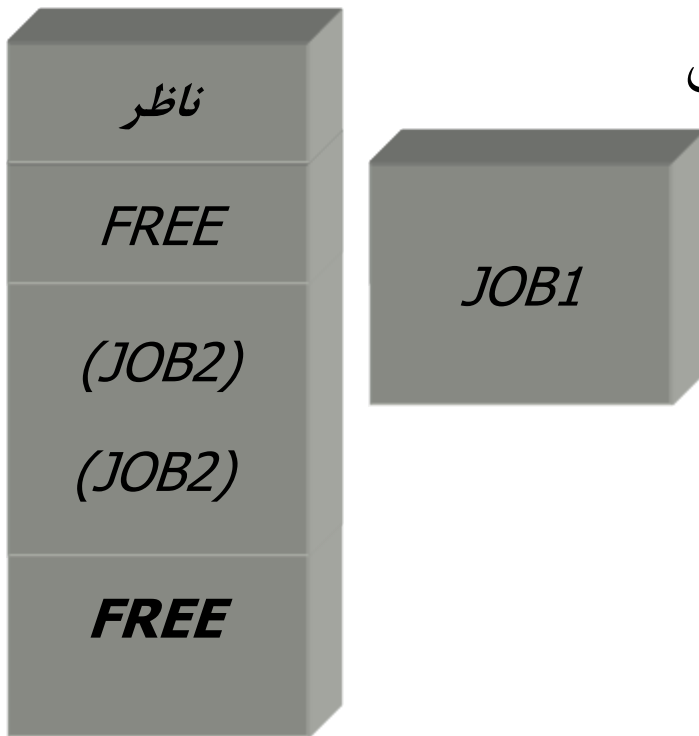
- چون Job3 از Job2 کوچکتر است مقداری از Job2 بر روی دیسک ذخیره میشود و Job3 از محل ۵۰۰۰ بار میشود.



(c)

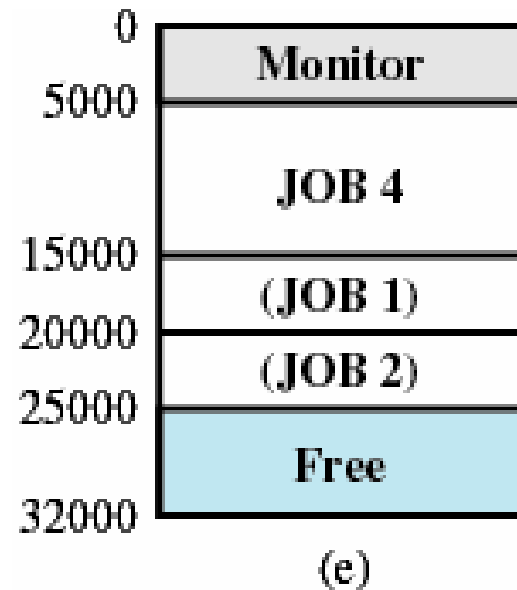
مثالی برای درک عملکرد CTSS :

- Job1 بزرگتر از Job3 و کوچکتر از Job2 است، کل Job2 و مقداری از Job2 روی دیسک ذخیره میشود و Job1 از محل ۵۰۰۰ بار میشود.



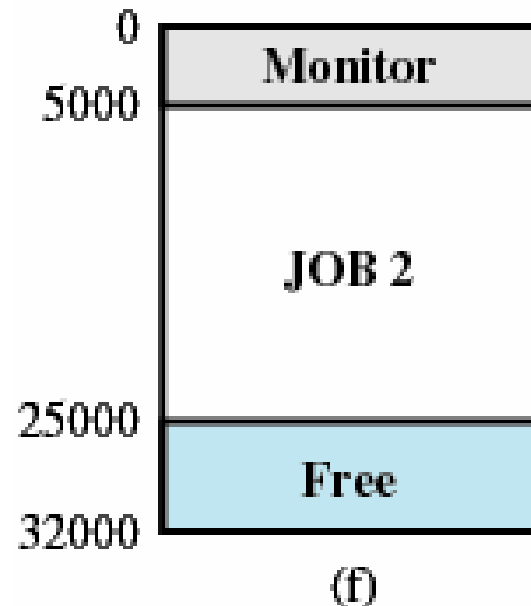
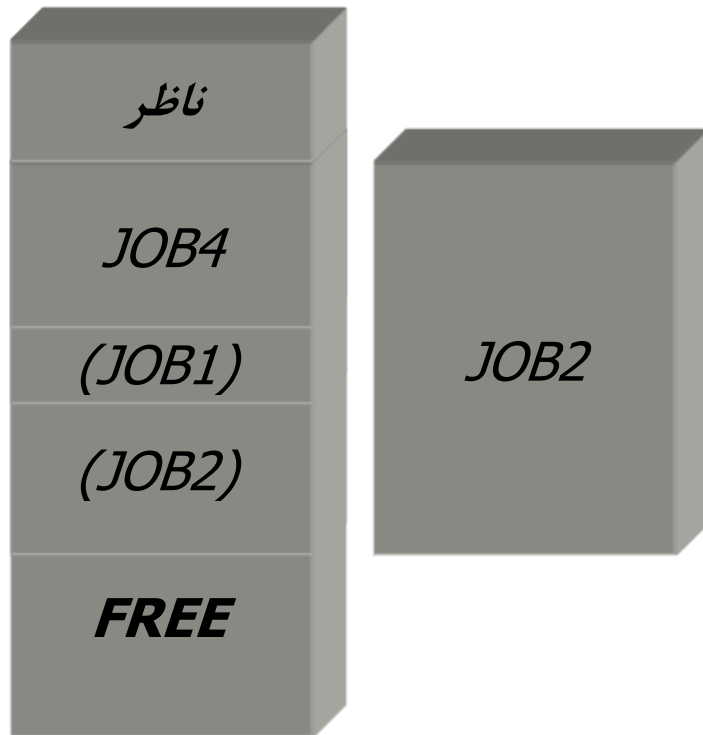
مثالی برای درک عملکرد CTSS :

- چون Job4 کوچکتر از Job1، مقداری از Job1 روی دیسک ذخیره میشود و کل Job4 از محل ۵۰۰۰ به بعد بار می شود



مثالی برای درک عملکرد CTSS :

- در این مرحله کل Job4 و بخشی از Job1 که در حافظه قرار دارد بر روی دیسک نوشته میشود و کل Job2 بار میشود.



دستاوردهای اصلی

پنج دستاورد توسعه سیستم عامل:

- ۱- فرآیندها.
- ۲- مدیریت حافظه.
- ۳- حفاظت اطلاعات و ایمنی.
- ۴- زمانبندی و مدیریت منابع.
- ۵- ساختار سیستم.

سه محور برای ایجاد و توسعه سیستم کامپیوتر:

۱- چند برنامگی:

برای مشغول نگهداشتن هم زمان پردازنده و ورودی/خروجی برای کارایی بیشتر.

۲- اشتراک زمانی:

آماده پاسخگویی هم زمان به چند کاربر.

۳- سیستم های تراکنش بلا درنگ:

تعدادی از کاربران در حال وارد کردن درخواستها یا تغییرات خود روی بانک اطلاعاتی باشند.

- هدف اشتراک زمانی و سیستم تراکنش بلا درنگ : حداقل زمان پاسخ گویی
- تفاوت اشتراک زمانی و سیستم تراکنش بلا درنگ : سیستم تراکنش بلا درنگ محدود به دو یا سه کاربرد ولی اشتراک زمانی کاربرد های مختلف دارد.

علت بروز خطاها در سیستم نرم افزاری:

- **همگام سازی نامناسب** : اغلب یک روال منتظر رسیدن علامتی از طرف روال دیگر است. طراحی نامناسب راهکار علامت دهی میتواند موجب از دست رفتن علامت ها شود.
- **شکست در انحصار متقابل**: در هر زمان تنها یک فرایند به یک منبع دسترسی داشته باشد، در غیر این صورت موجب بروز خطا خواهد شد.
- **عملکرد غیر قطعی برنامه**: به دلیل دسترسی برنامه های مختلف به حافظه ممکن است برنامه در نتیجه یکدیگر اثر بگذارند.
- **بن بست ها**: ممکن ۲ یا چند برنامه بخاطر یکدیگر معطل باشند. مثلاً دو برنامه هر کدام یک منبع را کنترل کند که دیگری به آن نیاز دارد.

فرآیندها

- یک مفهوم بنیادی در سیستم عامل است.
- واژه فرایند عمومی تر از واژه کار است،
- یک برنامه در حال اجرا است.
- موجودیتی که می توان به یک پردازنده نسبت داده شود و روی پردازنده اجرا شود.
- واحدی از فعالیت که بوسیله دنباله ای از اجرا، حالت موجود و مجموعه ای از منابع سیستم که به آن مربوط شده است مشخص میگردد.

اجزاء یک فرآیند:

✓ هر فرایند ۳ جزء دارد:

۱- یک برنامه قابل اجرا.

۲- داده های مورد نیاز این برنامه.

۳- متن یا وضعیت اجرای آن برنامه.

(این جزء از همه مهمتر است، و شامل تمام اطلاعاتی است که سیستم عامل برای مدیریت فرآیندها و اجرای دوباره آنها احتیاج دارد.)

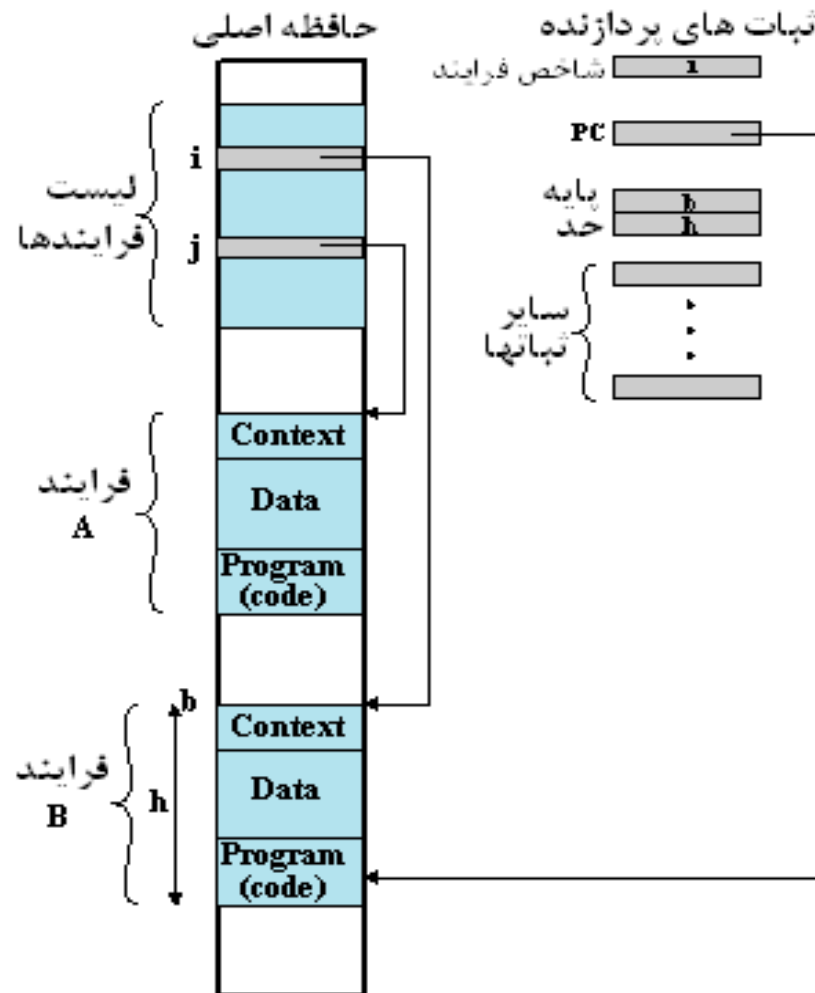
پیاده سازی متداول فرایند:

- در حافظه اصلی یک بلوک از حافظه برای نگهداری برنامه و داده ها و متن برنامه تخصیص داده میشود.
- هر فرایند در فهرست فرایند ها که توسط سیستم عامل ایجاد و نگهداری میشود ثبت شده است.
- در فهرست برای هر فرایند یک مدخل (شامل اشاره گری به بلوک فرایند در حافظه) وجود دارد.
- ثبات شاخص فرایند، حاوی شاخص عنصری از فهرست است که اکنون پردازنده را کنترل می کند.

پیاده سازی متداول فرایند:

- PC یا شمارنده برنامه به دستورالعمل بعدی اشاره میکند.
- ۲ ثبات پایه و حد ناحیه ای از حافظه که توسط فرایند اشغال شده را مشخص می کنند.
- شمارنده برنامه و تمام مراجعات به داده هانسبت به محتویات ثبات پایه تفسیر می شود و نباید از ثبات حد بیشتر باشد.
- از دخالت بین فرایندها جلوگیری می کنند.
- امکان دارد بروز وقفه حین اجرای فرایند B موجب توقف B و اجرای A شود.

پیاده سازی متداول فرایند:



سیستم عامل و مدیریت حافظه :

- جداسازی فرایندها : OS باید از مداخله فرایندها در داده های یکدیگر جلوگیری کند.
- تخصیص و مدیریت خودکار: در صورت نیاز باید به طور پویا به برنامه جا اختصاص داده شود.
- حمایت از برنامه سازی مؤلفه ای : برنامه نویس باید قادر به ایجاد حذف و تغییر اندازه مؤلفه ها باشد.
- حفاظت و کنترل دسترسی: OS باید اشتراک فرایندها به داده ها و حافظه را کنترل کند.
- حافظه دراز مدت : برای نگهداری داده ها و فرایندها به مدت طولانی نیاز به حافظه دراز مدت است.

حافظه مجازی:

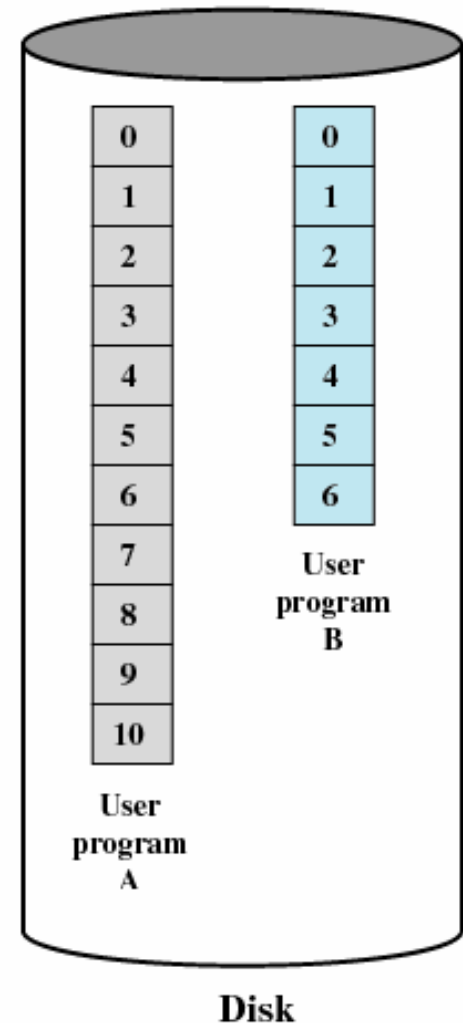
- حافظه مجازی: اجازه میدهد برنامه ها حافظه را از نقطه نظر منطقی و بدون توجه به اندازه فیزیکی حافظه آدرس دهی کنند.
- در سیستم پرونده و حافظه مجازی، اطلاعات میتوانند برای مدت طولانی ذخیره شوند. اطلاعات در اشیائی به نام پرونده ذخیره میشوند.

صفحه بندی:

-
- هر فرایند دارای تعدادی بلوک با طول ثابت به نام صفحه میباشد.
 - آدرس مجازی شامل شماره صفحه و یک انحراف در صفحه است.
 - هر صفحه ممکن است در هر جای حافظه اصلی قرار بگیرد.
 - سیستم صفحه بندی یک نگاشت پویا بین آدرس مجازی و آدرس فیزیکی در حافظه اصلی بوجود می آورد.

حافظه مجازی:

Secondary memory (disk) can hold many fixed-length pages. A User program consists of some Number of pages. Pages for all Program plus the operating system Are on disk, as are files



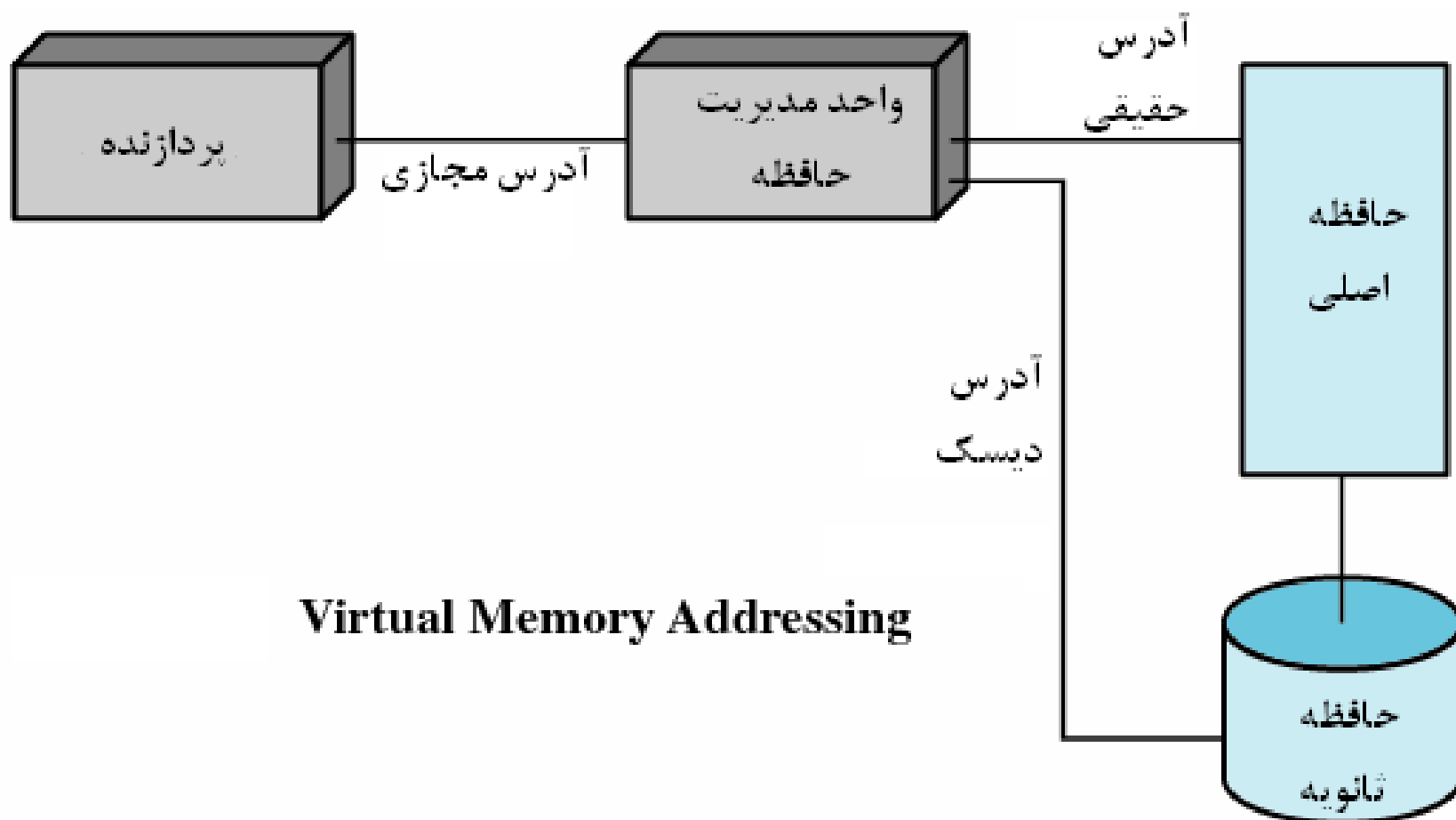
حافظه مجازی:

***Main memory consists of a
Number of fixed-length frames,
Each equal to the size of a page.
For a program to execute, some
Or all of it's pages must be in main memory***

A.1			
	A.0	A.2	
	A.5		
B.0	B.1	B.2	B.3
		A.7	
	A.9		
		A.8	
	B.5	B.6	

Main Memory

آدرس دهی حافظه مجازی:



حفاظت از اطلاعات و ایمنی:

- با رشد سیستم های اشتراک زمانی و شبکه نیاز به حفاظت اطلاعات بیشتر شد. بخش اعظم این کار توسط سیستم عامل انجام میشود. و به ۳ گروه زیر تقسیم میشود.

• **کنترل دسترسی :** تنظیم کنترل دسترسی کاربران به داده ها و منابع

• **کنترل جریان اطلاعات :** تنظیم جریان داده ها در داخل سیستم و تحویل آن به کاربر

• **گواهی:** اثبات اینکه مراحل بالا به درستی انجام شده.

زمانبندی و مدیریت منابع:

- یکی از وظایف سیستم عامل مدیریت منابع و زمانبندی استفاده از آنها توسط فرایندها است.
- سیاست تخصیص منابع به فرایندها باید ۳ شرط زیر را برآورده سازد:
 - **انصاف:** همه فرایندهای خواستار یک منبع باید حق دستیابی یکسان داشته باشند.
 - **حساسیت در مقابل تفاوت ها:** ممکن است نیاز باشد سیستم عامل بین کارهای با کلاسهای مختلف تفاوت بگذارد.
 - **کارایی:** باید به گونه باشد که توان عملیاتی حداکثر، حداقل زمان پاسخ، و در مورد سیستم های اشتراکی حداکثر کاربران را حمایت کند.

اجزای اصلی سیستم عامل برای زمانبندی و تخصیص منابع:

- **صف کوتاه مدت:** شامل فرایندهایی که در حافظه اصلی قرار دارند و در انتظار توزیع وقت پردازنده برای اجرا هستند.

- **صف دراز مدت:** فهرستی از کارهای جدید که برای اجرا ابتدا باید توسط پردازنده به صف کوتاه مدت منتقل شوند.

- **صف دستگاه I/O:** فرایندهایی که متقاضی استفاده از یک دستگاه I/O هستند در صف آن دستگاه قرار میگیرند.

ساختار سیستم:

• میتوان سیستم عامل را به صورت چند سطح در نظر گرفت که هر سطح زیر مجموعه ای از اعمال را نشان میدهد. سطوح پایین تر اعمال ابتدایی تر را انجام میدهند.

• ۴ سطح اول مربوط به سخت افزار میشود و جزء سیستم عامل نیستند.

ساختار سیستم:

• سطح ۱:

- مدارهای الکترونیکی
- ثبات ها، دروازه ها ، گذرگاه ها
- عملیات: پاک کردن، انتقال، فعال کردن، مکمل گیری

• سطح ۲ :

- مجموعه دستورالعملها
- پشته ارزیابی، مفسر ریز برنامه داده های عددی و برداری
- بار کردن، ذخیره کردن، جمع کردن، تفریق کردن، انشعاب

ساختار سیستم:

- سطح ۳ :

- رویه ها

- رویه ها ، پشته فراخوانی

- علامت زدت پشته، فراخوانی ، بازگشت

- سطح ۴ :

- وقفه ها

- برنامه گرداننده وقفه

- نقاب زدن، رفع نقاب، تلاش کردن

ساختار سیستم:

• سطح ۵ :

- فرایندهای اولیه
- فرایندهای اولیه، راهنماها، فهرست اولیه
- فهرست آماده

• سطح ۶ :

- حافظه ثانویه محلی
- بلوکهای داده، کانالهای دستگاه
- خواندن، نوشتن، تخصیص، رها کردن

ساختار سیستم:

- سطح ۷ :

- حافظه مجازی
- قطعه ها ، صفحه ها
- خواندن، نوشتن، واکشی

- سطح ۸ :

- ارتباطات
- لوله ها
- ایجاد، تخریب، باز کردن، خواندن، نوشتن، بستن

ساختار سیستم:

• سطح ۹ :

- سیستم پرونده ها
- پرونده ها
- ایجاد، تخریب، باز کردن، خواندن، نوشتن، بستن

• سطح ۱۰ :

- دستگاه ها
- دستگاه های خارجی مثل چاپگر، صفحات نمایش، ...
- باز کردن، خواندن، نوشتن، بستن

ساختار سیستم:

• سطح ۱۱ :

- فهرست راهنما ها
- فهرست راهنما ها
- ایجاد، تخریب، اتصال، انفصال، جستجو ، فهرست کردن

• سطح ۱۲ :

- فرایندهای کاربر
- محیط برنامه نویسی کاربر
- خروج، کشتن ، تعلیق، ازسرگیری

ساختار سیستم:

- سطح ۱۳ :

- پوسته

- محیط برنامه نویسی کاربر

- دستورات به زبان پوسته

ویژگی های سیستم عامل های جدید:

- معماری ریز هسته.
- چند نخه.
- چند پردازشی متقارن.
- سیستم های عامل توزیعی.
- طراحی شیء گرا.

معماری ریز هسته

- در این معماری تنها تعداد کمی توابع ضروری در هسته قرار می گیرند.
- این دستورالعمل ها عبارتند از:
 - زمان بندی پایه ای
 - دسترسی به فضای آدرس
 - ارتباط بین فرایندها

رویکرد ریز هسته:

- ۱- پیاده سازی را ساده می کند.
- ۲- موجب انعطاف می گردد.
- ۳- محیط توزیعی کاملاً سازگار.

چند نخ

- فرآیندها به نخ هایی تقسیم می شوند که می توانند به طور هم زمان اجرا شوند
- در فرآیندهایی که چند وظیفه اصلی مستقل را انجام می دهند مفید است.
- نخ یک واحد کاریست که میتواند وقت پردازنده را به خود اختصاص دهد.
- فرایند : مجموعه یک یا چند نخ و منابع سیستمی تخصیص داده شده (مشابه یک برنامه حال اجرا)

چند پردازی متقارن

- ۱- پردازنده متعددی وجود دارد.
- ۲- از امکانات ورودی / خروجی و حافظه اصلی به طور مشترک استفاده می کنند.
- ۳- تمام پردازنده ها اعمال یکسانی را می توانند انجام دهند.
- ۴- هر پردازنده میتواند فرایندی مجزا را اجرا کند.

امتیازات چند پردازی متقارن نسبت به معماری تک پردازنده:

- کارایی: قرار دادن ترتیب کارها به نحوی که به موازات هم انجام شوند.
- دسترسی پذیری: از آنجا که تمام پردازنده ها یک عمل را انجام می دهد خرابی در یکی از آنها موجب توقف ماشین نمی شود.
- رشد: با اضافه کردن پردازنده.
- مقیاس پذیری: قیمت ها متناسب با تعداد پردازنده ها باشد.

سیستم های عامل توزیعی

تصور یک فضای حافظه اصلی واحد و یک فضای حافظه ثانوی واحد و دیگر امکاناتی که دسترسی به آنها یکنواخت شده است را ارائه می کند.

طراحی شیء گرا

- افزودن ملحقات مولفه ای به یک هسته کوچک.
- محیا ساختن ابزار های توزیعی و سیستم عامل های توزیعی را آماده میکند.