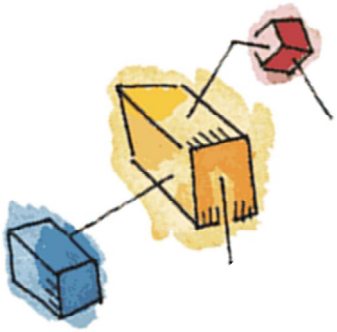
به نام خدا

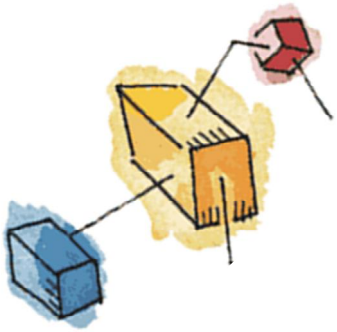
فصل دوم
نگاهی کلی به سیستم عامل

Operating System Overview

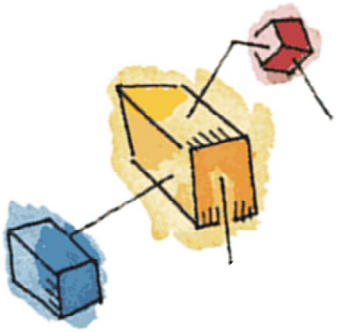


سرفصل کلی مطالب

i. اهداف و وظایف سیستم عامل
ii. تکامل تدریجی سیستم عامل
iii. دستاوردهای اصلی توسعه سیستم عامل
iv. ویژگی های سیستم عامل های جدید



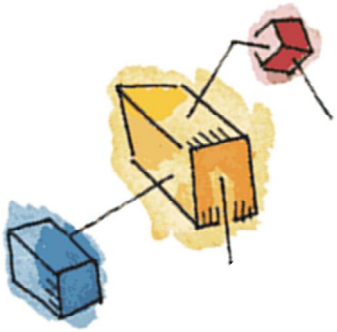
أ. اهداف و وظایف سیستم عامل



سیستم عامل

- سیستم عامل یک برنامه است که اجرای برنامه های کاربردی را کنترل می کند.

- سیستم عامل بصورت یک رابط میان سخت افزار و برنامه های کاربردی عمل می کند.



اهداف سیستم عامل

- سهولت
  - استفاده از کامپیوتر را آسان می کند.
- کارآمدی
  - موجب استفاده کارآمدتر از منابع سیستم می شود.
- قابلیت رشد و تکامل
  - باید به گونه ای باشد که قابلیت رشد و تکامل داشته باشد.
  - اجازه ایجاد و توسعه موثر توابع سیستمی جدید را بدهد.

لایه های یک سیستم کامپیوتری

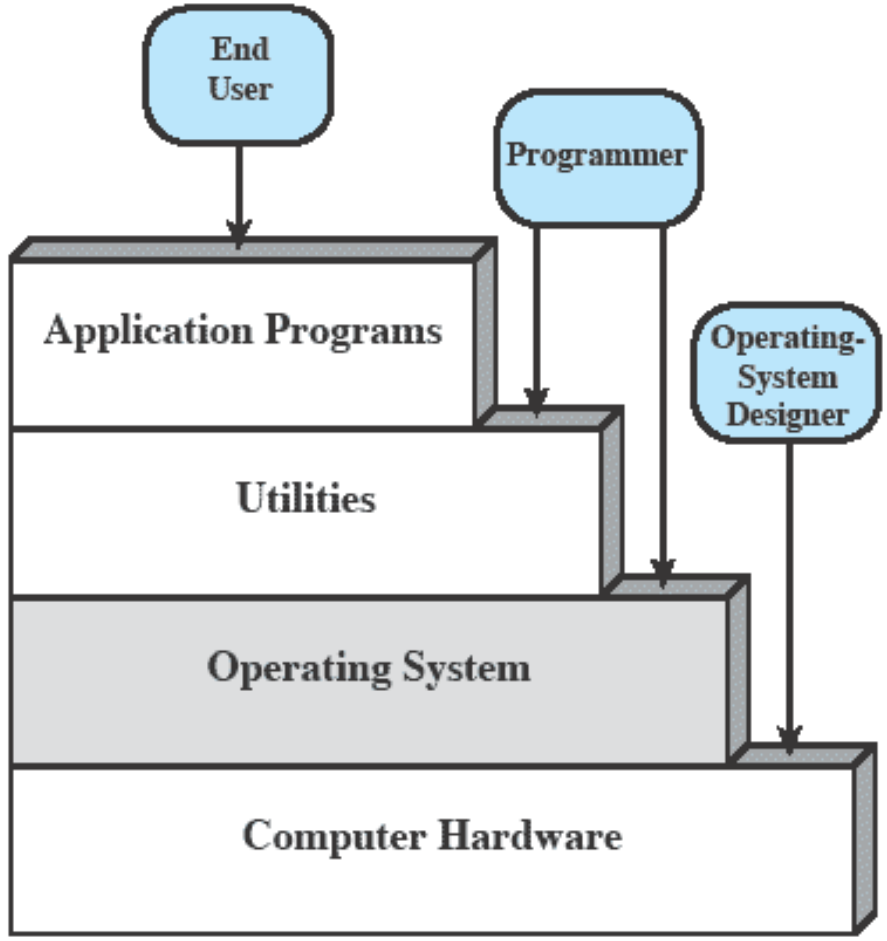
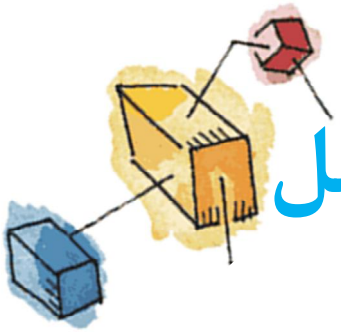
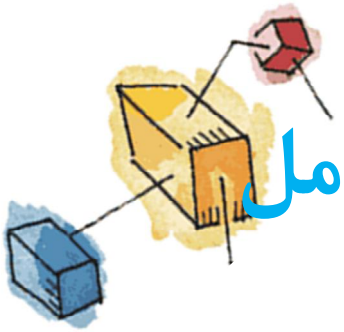


Figure 2.1 Layers and Views of a Computer System



سرویس های ارائه شده توسط سیستم عامل

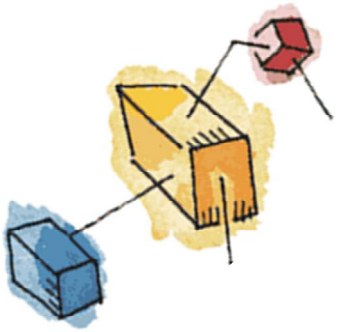
- **توسعه برنامه**
  - ارائه ابزارها و امکاناتی برای توسعه برنامه ها
  - همچون ویراستارها و اشکال زداها
- **اجرای برنامه (با انجام زمان بندی و ....)**
- **دسترسی به دستگاه های ورودی/خروجی (I/O)**
- **دسترسی کنترل شده به فایل ها**
  - راهکارهای حفاظتی برای دسترسی به سیستم فایل
- **دسترسی به سیستم**
  - کنترل دسترسی کاربران به سیستم (جلوگیری از دسترسی های غیرمجاز)



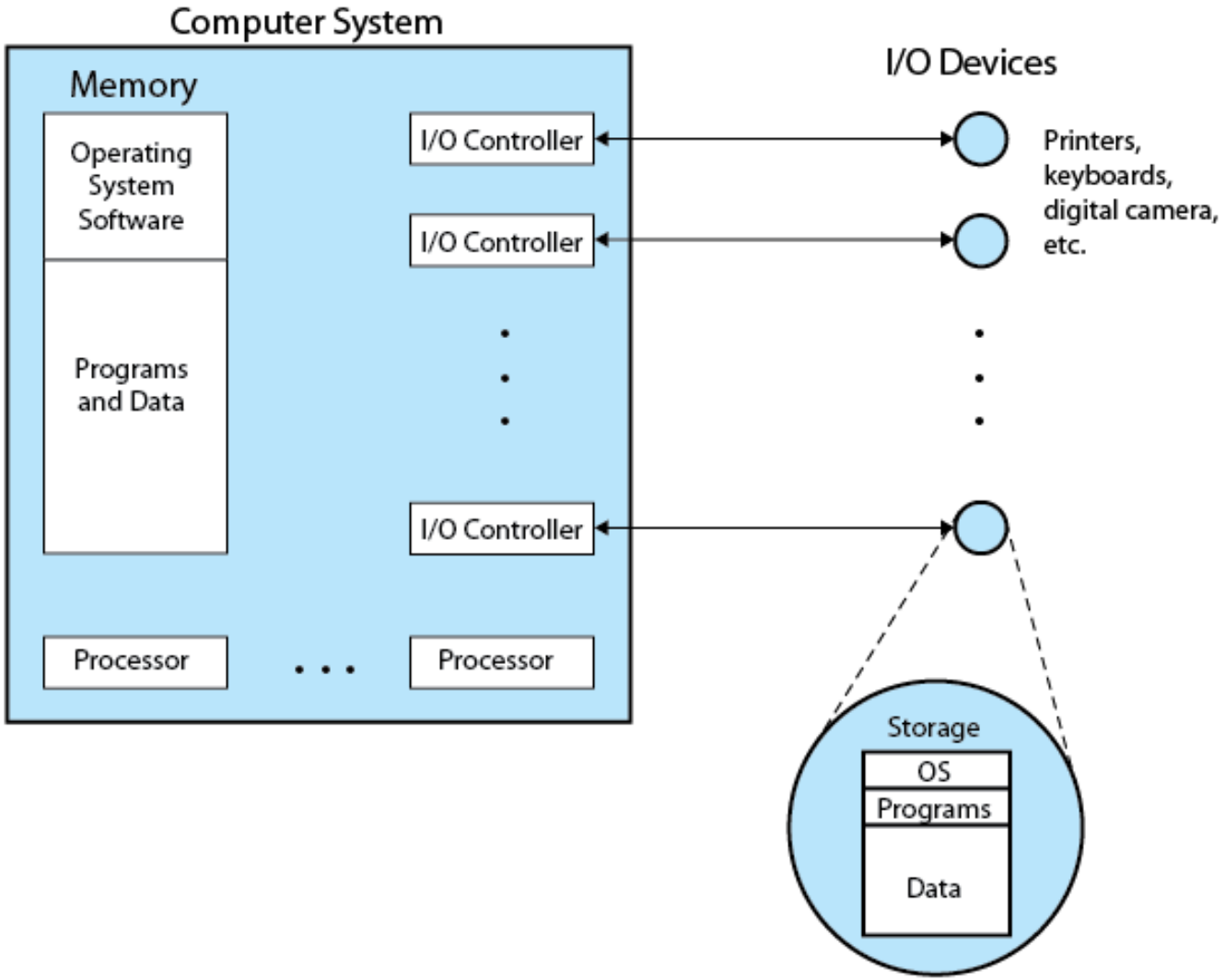
سرویس های ارائه شده توسط سیستم عامل

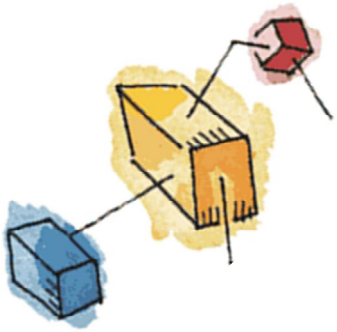
- **کشف و پاسخ خطا**
  - شناسایی خطاهای سخت افزاری
  - شناسایی خطاهای نرم افزاری
  - عکس العمل و پاسخگویی

- **حسابداری**
  - جمع آوری آمار استفاده از منابع
  - نظارت بر کارایی سیستم
  - این آمار برای بهبودها و ارتقاء های آینده سیستم استفاده می شود.
  - همچنین این آمارها برای عملیات صورت حساب نیز به کار می رود.
  -



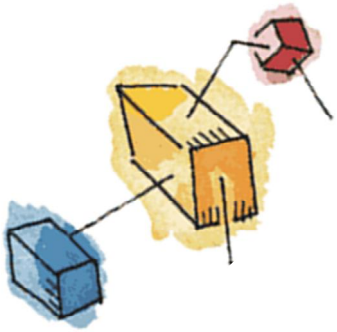
سیستم عامل به عنوان مدیر منابع



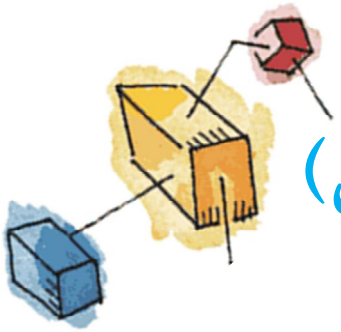


تکامل سیستم عامل

- یک سیستم عامل ممکن است به دلایل زیر در طول زمان تغییر کند:
  - ارتقاء سخت افزار و یا ظهور انواع جدید سخت افزار
  - ارائه خدمات جدید
  - رفع خطاهای کشف شده در سیستم عامل



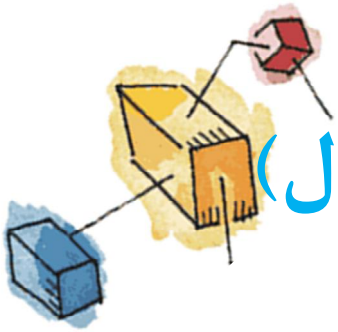
ii. تکامل تدریجی سیستم عامل



تکامل سیستم عامل: ۱) پردازش ردیفی (سریال)

- اواخر ۱۹۴۰ تا اواسط ۱۹۵۰
- ابتدا سیستم عاملی وجود نداشت.
- کاربران یکی یکی به صورت ردیفی به کامپیوتر دسترسی داشتند.
- برنامه نویس مستقیما با سخت افزار در تراکنش بود.

- ماشین از طریق یک میزفرمان (شامل چراغ های نمایش، کلیدها، نوعی دستگاه ورودی و چاپگر) اجرا می شد.
- برنامه ها به زبان ماشین بر روی کارت نوشته می شد و به دستگاه ورودی (کارت خوان) بار می شد.
- اگر خطایی رخ می داد، برنامه متوقف می شد و شرایط خطا توسط چراغ ها اعلام می شد و برنامه نویس سعی می کرد با بررسی ثبات های پردازنده و حافظه اصلی، علت خطا را تعیین کند.
- با تکمیل طبیعی برنامه (یعنی خاتمه بدون خطا) خروجی بر روی چاپگر ظاهر می شد.

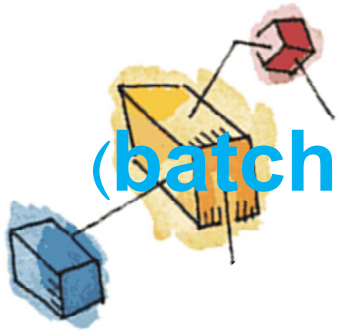


تکامل سیستم عامل: ۱) پردازش ردیفی (سریال)

- **زمانبندی:** هر کاربر باید از برگه های نوبت گیری استفاده می کرد (معمولا مضربی از زمان های نیم ساعته). در صورت کامل نشدن در موقع مقرر، برنامه خاتمه می یافت تا بعداً دوباره از ابتدا اجرا شود. اگر زودتر از موعد، اجرای برنامه متوقف می شد، وقت کامپیوتر هدر می رفت.

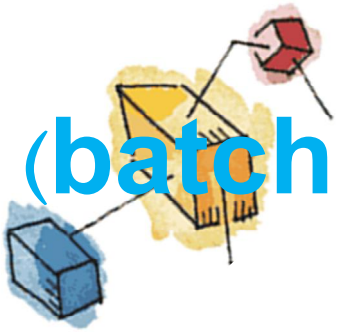
- **برپا کردن برنامه (setup):** هر برنامه که کار (job) نامیده می شد، ممکن بود شامل بار کردن مترجم و کد منبع به حافظه، ذخیره سازی برنامهٔ ترجمه شده، بارگذاری و اتصال آن با توابع متداول باشد. هر کدام از این گام ها نیازمند قراردادن و برداشتن نوارها و کارت ها بود.

  – در صورت بروز خطا، کاربر کار را از اول شروع می کرد. بنابراین وقت قابل ملاحظه ای برای نصب و شروع به کار برنامه صرف می شد.



تکامل سیستم عامل: ۲) پردازش دسته ای ساده (batch)

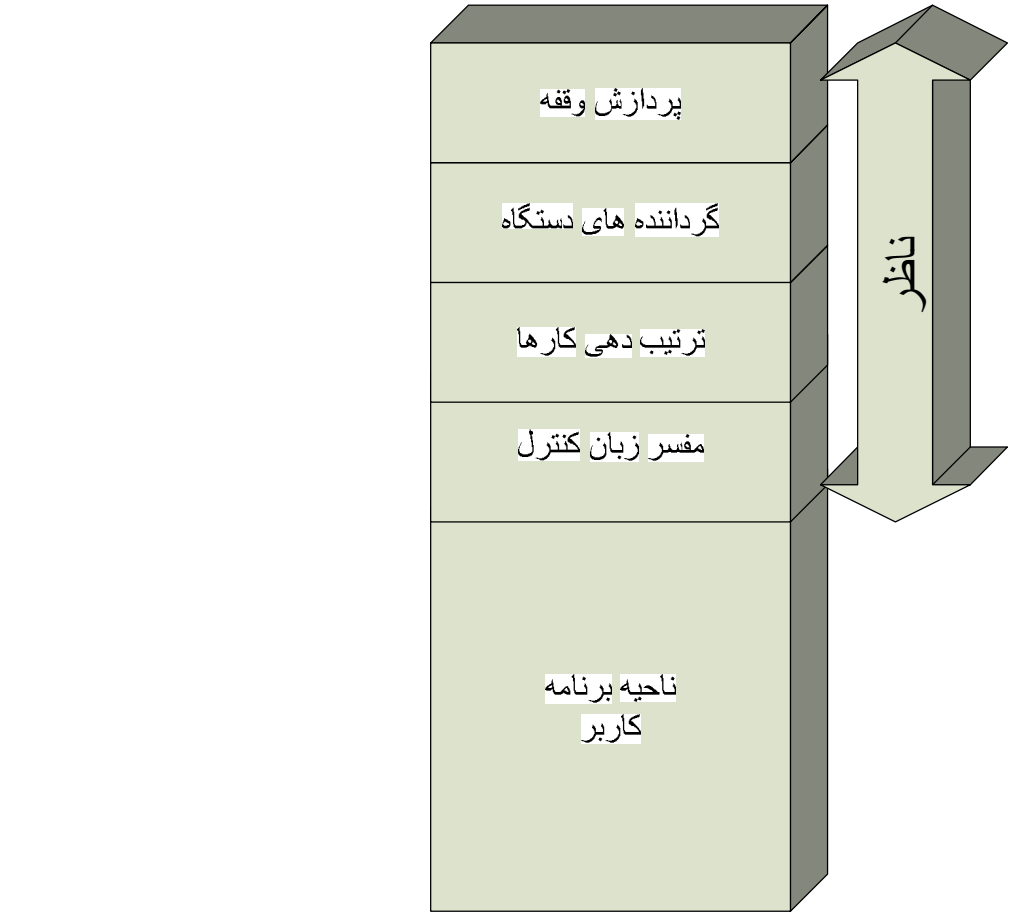
- کامپیوترهای اولیه بسیار گران بودند و بنابراین حداکثر استفاده از کامپیوتر مهم بود.
- زمانی که به خاطر زمانبندی و نصب در سیستم های ردیفی هدر می رفت قابل قبول نبود.
- سیستم عامل دسته ای در اواسط دهه ۱۹۵۰ ارائه شد.
- در آن، از برنامه ای به نام ناظر (Monitor) استفاده شد.
- برنامه ناظر، توالی رویدادها را کنترل می کرد.
- کاربر دسترسی مستقیم به ماشین نداشت. کاربر برنامه را بر روی کارت به متصدی می داد و متصدی کارت ها را به طور ردیفی دسته کرده و در دستگاه نوار خوان قرار می داد تا مورد استفاده ناظر قرار بگیرد.
- هر برنامه به نوعی ساخته می شد که با تکمیل پردازش، به برنامه ناظر پرش می کرد و برنامه ناظر به طور خودکار، برنامه بعدی را لود می کرد.

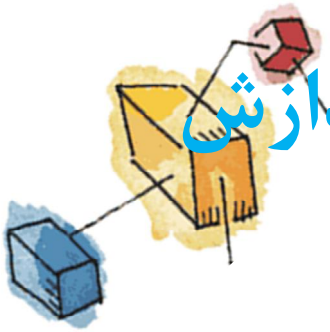


تکامل سیستم عامل: ۲) پردازش دسته ای ساده (batch)

- چون ناظر اکثر عملیات را انجام می دهد، نیاز است که بخش اعظمی از آن در حافظه باشد. این بخش را ناظر ماندگار می گویند.
- بقیه برنامه ناظر شامل برنامه های سودمند و توابع عمومی و مشترکی است که در ابتدای کارهایی (برنامه هایی) که به آنها نیاز دارند در حافظه لود می شود.
- ناظر کارها را یکی یکی از دستگاه ورودی (معمولا دستگاه کارت خوان) می خواند. آن کار در ناحیه کاربر در حافظه قرار گرفته و کنترل به آن منتقل می شود (یعنی پردازنده مشغول اجرای دستورات آن می شود).
- با تکمیل کار، کنترل به ناظر بر می گردد (یعنی پردازنده مشغول اجرای دستورات برنامه ناظر می شود) تا برنامه ناظر بلافاصله کار بعدی را از ورودی بخواند.
- نتایج هرکار به دستگاه خروجی می رود تا تحویل کاربر مربوط به آن شود.

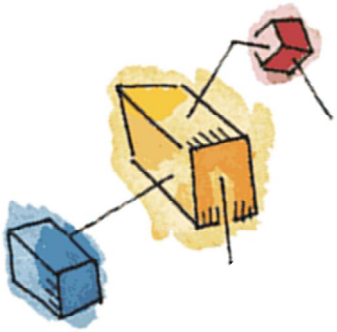
وضعیت حافظه برای ناظر ماندگار در حافظه





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 کند 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** (Privileged instructions)
  - دستورالعمل هایی که تنها توسط ناظر اجرا می شوند (مثل دستورات I/O). اگر برنامه کاربر بخواهد آنها را اجرا کند با خطا مواجه می شود و کنترل به ناظر منتقل می شود. برنامه کاربر باید از ناظر بخواهد تا آن دستور را برایش انجام دهد.
- **وقفه ها**
  - این خصوصیت به سیستم عامل انعطاف بیشتری می دهد.
  - سیستم عامل های اولیه این قابلیت را نداشتند.



حفاظت از حافظه اصلی

- برنامه های کاربر در حالت کاربر (User Mode) اجرا می شوند.
  - بعضی دستورالعمل ها نمی توانند اجرا شوند.
- ناظر در حالت سیستم (System Mode) اجرا می شود.
  - به حالت سیستم حالت هسته یا ممتاز نیز گفته می شود.
  - دستورالعمل های ممتاز در حالت ممتاز اجرا می شوند (مانند دستورات کار با ورودی/خروجی).
  - قسمت های محافظت شده از حافظه ممکن است در این حالت در دسترس باشند.

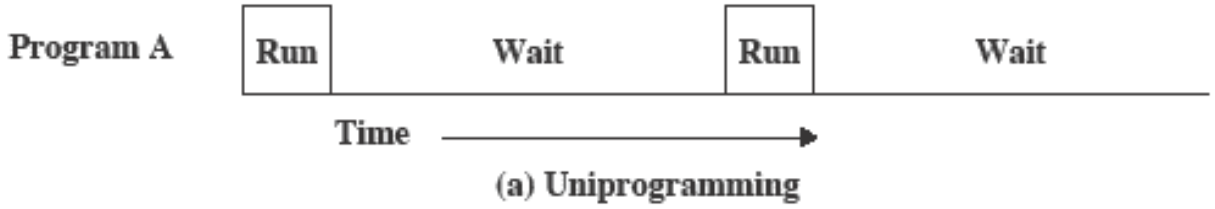
تکامل سیستم عامل: ۳) سیستم عامل چند برنامه ای دسته ای (Multiprogramming batch)

- 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 وCPU، باز هم پردازنده اکثر وقت خود را بیکار است.
- با هدف استفاده بیشتر و بهتر از وقت پردازنده، سیستم های چند برنامه ای دسته ای به جای سیستم های تک برنامه ای دسته ای (Uniprogramming batch) مورد استفاده قرار گرفتند.
- اگر ناحیه کاربر، چندین برنامه در حال اجرا را در خود داشته باشد، در حین اجرای عمل I/O برای یک برنامه، سیستم عامل می تواند برنامه دیگری را برای اجرا به پردازنده بسپارد.

- 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.

تک برنامه ای (Uniprogramming)

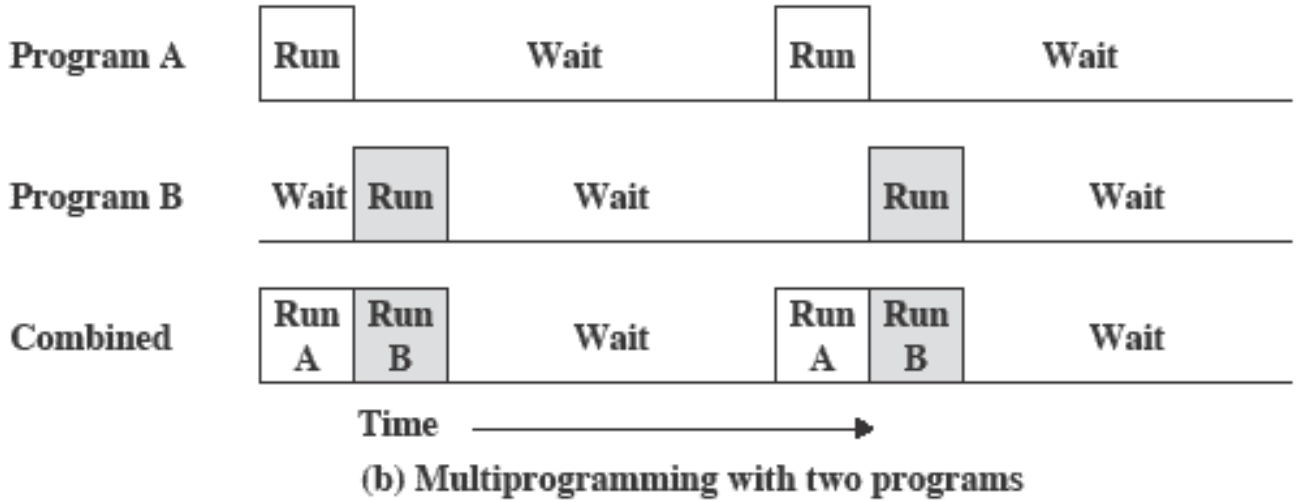
پردازنده باید منتظر بماند تا کار دستگاه I/O تمام شود، سپس به کارش ادامه دهد.



(a) Uniprogramming

چند برنامه ای (Multiprogramming)

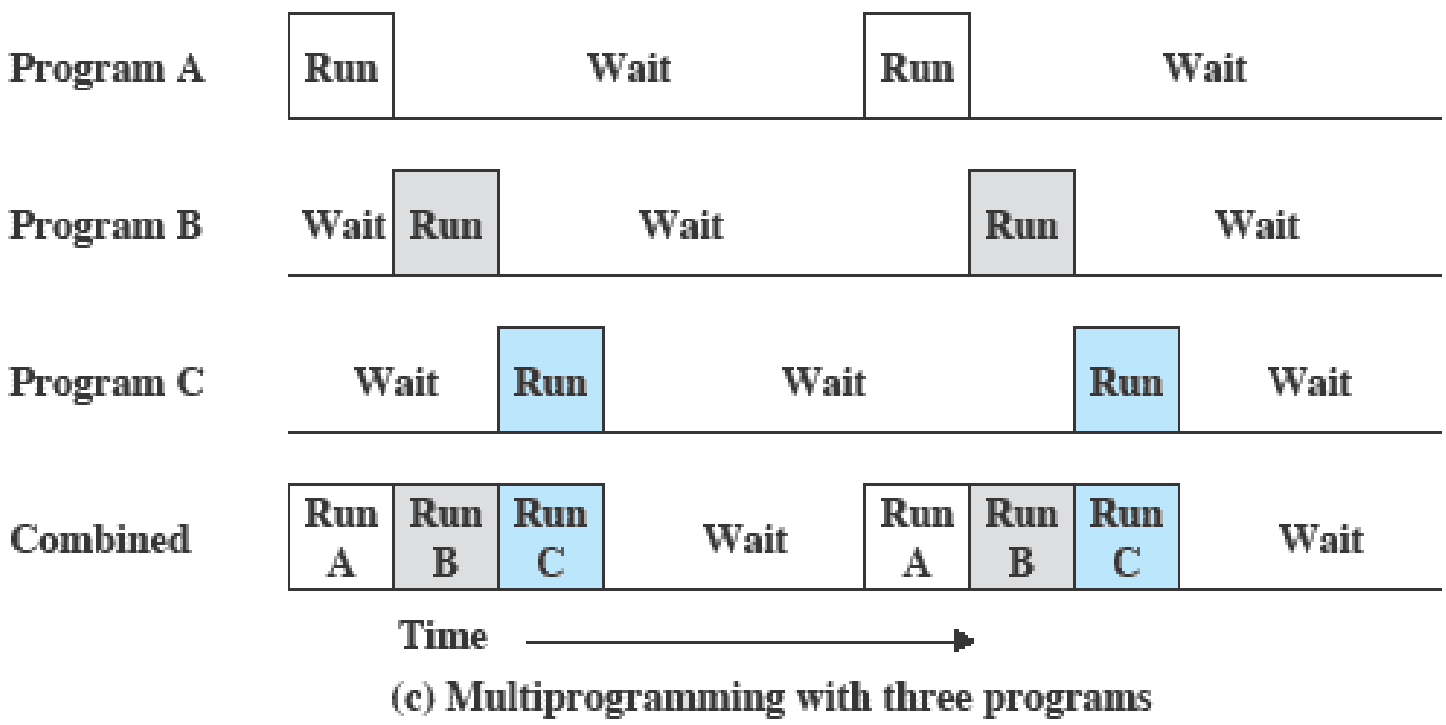
- هنگامی که کاری منتظر تکمیل ورودی/خروجی است، پردازنده مشغول اجرای کار دیگری می شود.
- چند برنامه ای با دو برنامه:



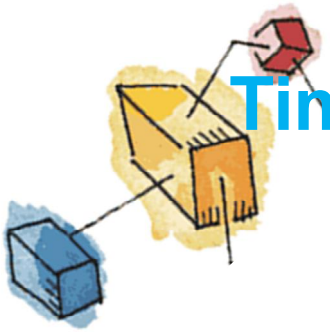
(b) Multiprogramming with two programs

چند برنامه ای (Multiprogramming)

چند برنامه ای با سه برنامه:

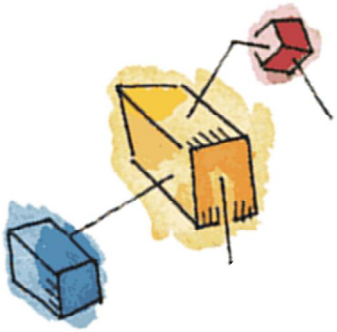


(c) Multiprogramming with three programs



تکامل سیستم عامل: ۴) اشتراک زمانی (Time Sharing)

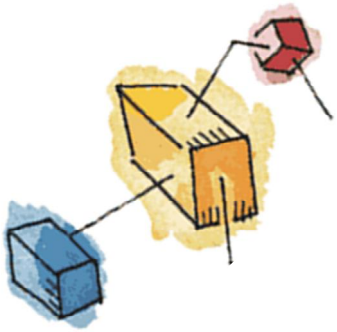
- برای بسیاری از کارها، وجود حالتی که کاربر مستقیما با کامپیوتر محاوره کند مطلوب است. لذا سیستم های اشتراک زمانی به وجود آمدند (دهه ۱۹۶۰).
- چند برنامگی امکان رسیدگی به چندین کار محاوره ای را می دهد.
- مبانی این روش، داشتن چندین کاربر بطور همزمان است که از طریق پایانه ها به سیستم دسترسی دارند.
- وقت پردازنده بین کاربران تقسیم می شود. در صورت وجود n کاربر، به هر یک از کاربران حدود 1/n وقت پردازنده می رسد (وقتی که صرفا صرف کاربران می شود).
- بین اجرای دوبرنامه متوالی، مدت کوتاهی سیستم عامل اجرا می شود.
- هدف اولیه و اصلی این سیستم ها کاهش زمان پاسخ به کاربران می باشد.



هدف سیستم های اشتراک زمانی

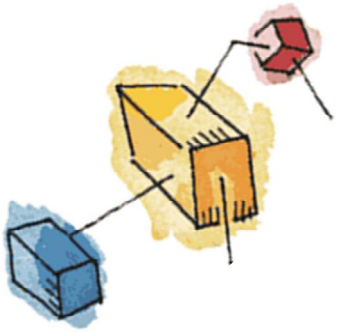
- نظر به کندی نسبی عکس العمل انسان نسبت به سرعت کامپیوتر، در سیستمی که خوب طراحی شده باشد، زمان پاسخ باید با زمان پاسخ یک کامپیوتر که به طور اختصاصی تنها برای یک کاربر است، نزدیک باشد.

- هدف اصلی سیستم های دسته ای چند برنامه ای، حداکثر استفاده از پردازنده بود. در حالیکه هدف اصلی سیستم های اشتراک زمانی، حداقل کردن زمان پاسخ به کاربر است.
- سیستم عامل CTSS (Compatible Time Sharing System) ، اولین سیستم اشتراک زمانی است و در سال ۱۹۶۱ در دانشگاه MIT توسعه یافت.



اولین سیستم اشتراک زمانی: CTSS (Compatible Time Sharing System)

- این سیستم روی ماشینی با ۳۲۰۰۰ کلمه حافظه ۳۲ 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 هم کار ناظر و هم مدیریت حافظه آسان تر می شد.
- در این سیستم در هر ۰/۲ ثانیه وقفه ای صادر می شد که موجب می شد پردازنده کنترل را به کار دیگری بدهد. بنابراین، در فواصل منظم، کار جاری متوقف و برنامه دیگری بار می شد.
- برای حفظ وضعیت کاربر قبلی و امکان از سرگیری آن، ابتدا برنامه ها و داده ها و وضعیت آن کاربر روی دیسک ذخیره می شد و سپس برنامه ها و داده های کاربر بعدی خوانده می شد. بعدا دوباره نوبت کاربر قبل می شد....



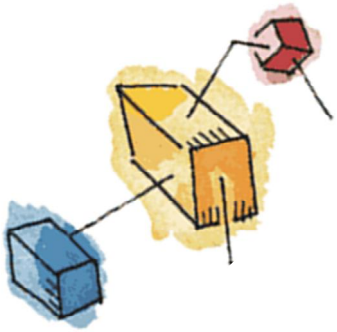
مثالی برای درک عملکرد CTSS

- فرض کنید چهار کاربر محاوره ای هر یک به حافظه هایی با اندازه های زیر نیاز دارند (بر اساس کلمه):

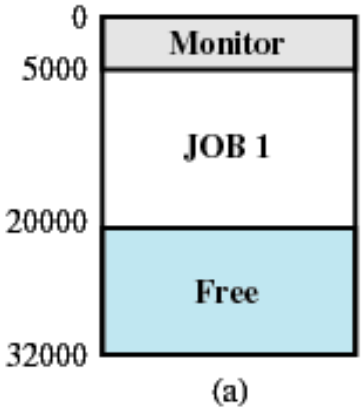
Job1: 15000 Job2: 20000

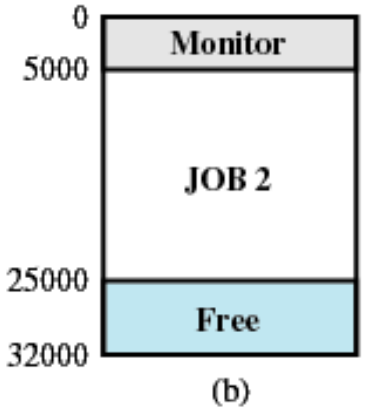
Job3: 5000 Job4: 10000

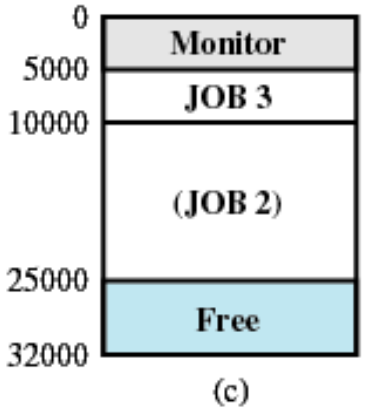
چون هنگام جابجایی کارها بر روی حافظه اصلی، مقداری از برنامه قبلی بر روی حافظه مانده و رونویسی نشده است (به علت متفاوت بودن اندازه برنامه ها)، زمان نوشتن مجدد برنامه کاهش می یابد.

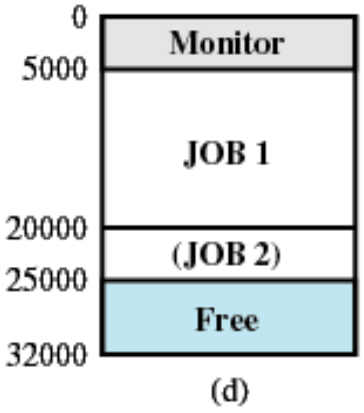


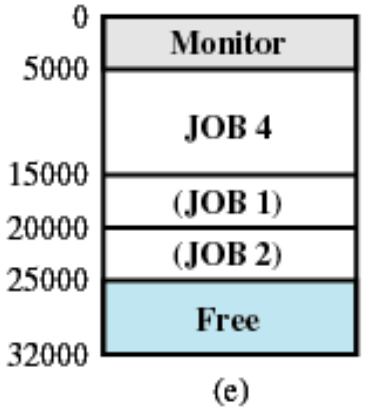
عمليات CTSS

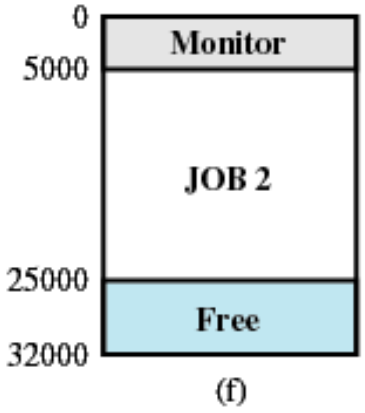

0
5000
20000
32000
Monitor
JOB 1
Free
(a)

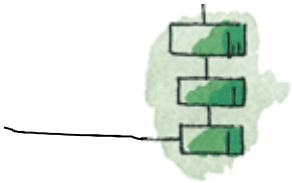

0
5000
25000
32000
Monitor
JOB 2
Free
(b)

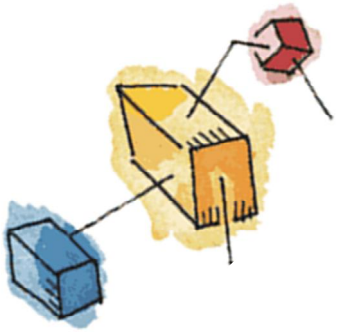

0
5000
10000
25000
32000
Monitor
JOB 3
(JOB 2)
Free
(c)


0
5000
20000
25000
32000
Monitor
JOB 1
(JOB 2)
Free
(d)


0
5000
15000
20000
25000
32000
Monitor
JOB 4
(JOB 1)
(JOB 2)
Free
(e)


0
5000
25000
32000
Monitor
JOB 2
Free
(f)





مسائل مطرح در سیستم های اشتراک زمانی

- این رویکرد در مقایسه با سیستم های اشتراک زمانی امروزی، بسیار ساده بود اما به هر حال کار می کرد.
- این سیستم تا حداکثر ۳۲ کاربر را پشتیبانی می کرد.
- سیستم های اشتراک زمانی و چندبرنامه ای باعث شدند مسائل جدیدی در رابطه با سیستم عامل مطرح شود.
  - اگر چندکار در حافظه هستند، باید از دسترسی یکدیگر محافظت شوند. همچنین ممکن است نیاز به جابجایی برنامه ها در حافظه باشد.
  - بر روی استفاده از منابعی مانند چاپگر و دستگاه های جانبی دیگر رقابت هست که باید مدیریت شود.
  - ....
  - ~~این موارد در فصل های بعدی کتاب بررسی خواهند شد.~~