سیستم عامل

Operating Systems

فرجیان

IASBS
1992-2012

فصل دوم:

نگاهی کلی به

سیستم عامل

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 کارامد 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.
- مجموعه ای از برنامه های سیستمی تهیه میشود
- **مهمترین برنامه سیستمی سیستم عامل است**، که به صورت میانجی بین سخت افزار و برنامه ساز است.

لایه ها یک سیستم کامپیوتری:

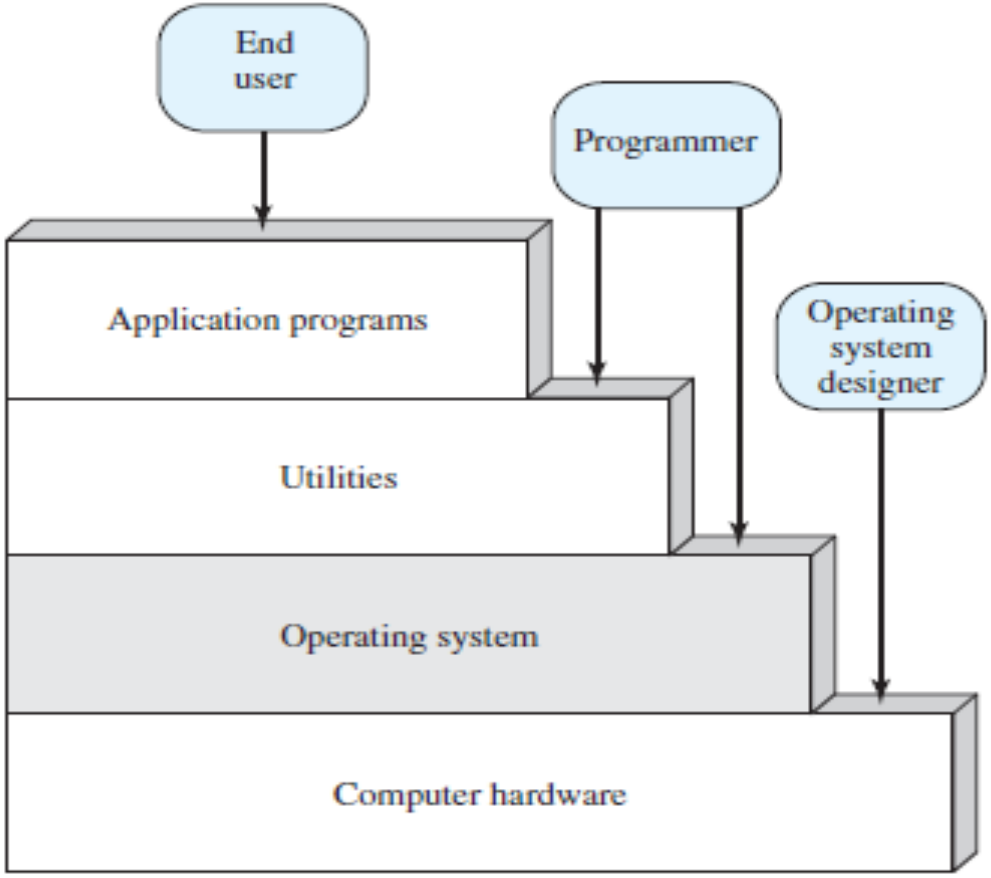


Figure 2.1 **Layers and Views of a Computer System**

خدمات سیستم عامل:

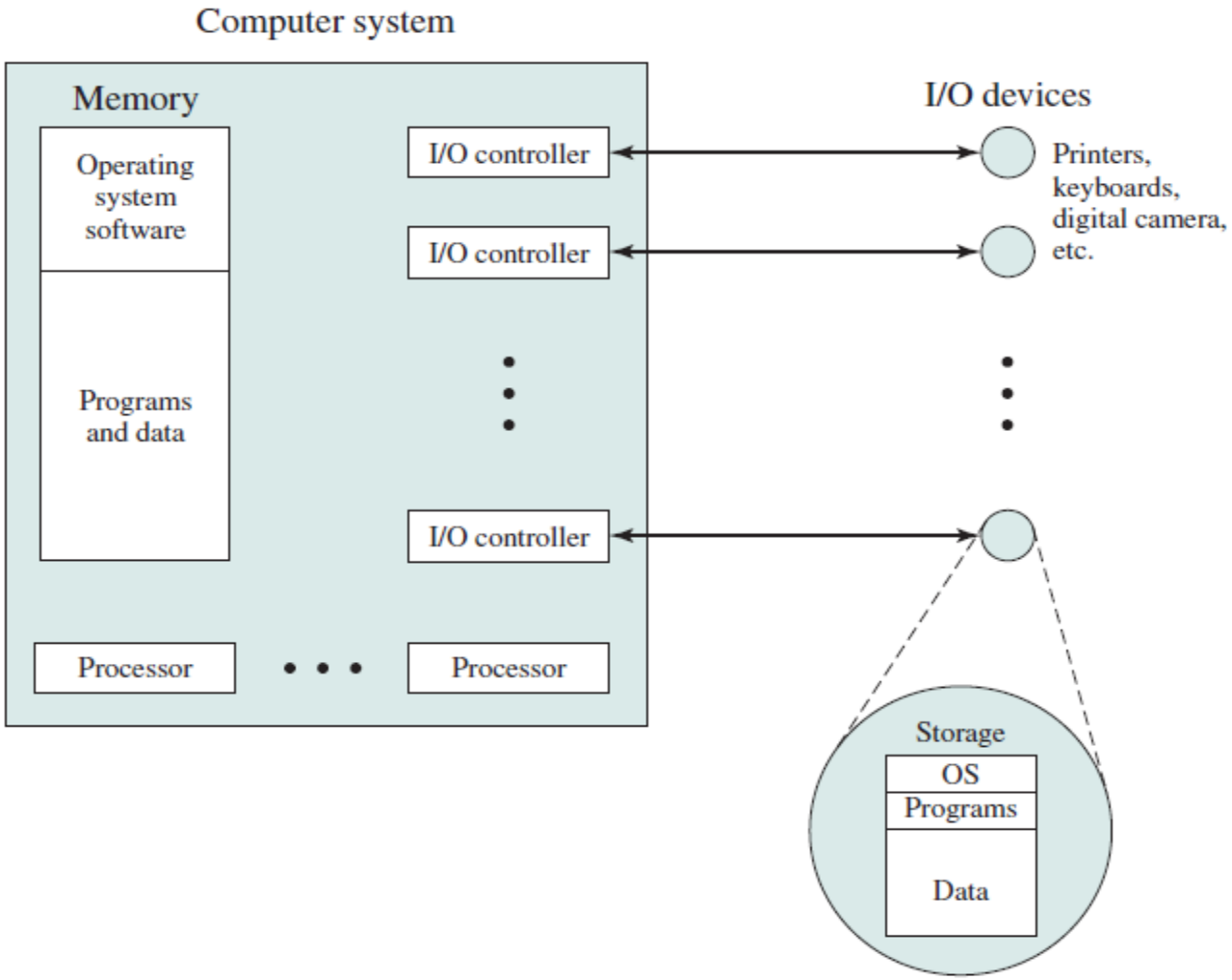
- **توسعه برنامه** :برنامه های سودمندی مثل editorsکه از طریق OS قابل دسترسند.
- **اجرای برنامه** : OSتمام مراحل اجرا (بارکردن داده ها و دستورات در حافظه، مقدار گذاری I/O) را زمان بندی میکند.

- **دسترسی به I/O** : فراهم کردن واسطی یکنواخت برای تمام دستگاه های I/O
- **کنترل دسترسی به پرونده ها** : ماهیت دستگاه (دیسک،نوار) ، ساختار داده ای محتوای پرونده 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 داده هاست.
- سیستم عامل مسئول مدیریت منابع است.
- سیستم عامل مانند سایر نرم افزارهاست
  - مانند سایر نرم افزار ها اجرا میشود.
- سیستم عامل مرتبا کنترل را به پردازنده واگذار می کند.

سیستم عامل به عنوان مدیر منابع:



وظایف سیستم عامل

- مدیریت حافظه .
- مدیریت فرآیند ها (زمانبندي و...)
- مدیریت ورود/خروج.
- مدیریت فایل ها.
- حفاظت خود از کاربر و حفاظت کاربران در مقابل هم.
- ثبت و نگهداری اشتباهات.
- نقل و انتقال اطلاعات بین کامپیوتر و دستگاهای جانبی.

IASBS
1992-2012

هسته:

- بخشی از سیستم عامل که در حافظه اصلی قرار دارد.

- هسته سیستم عامل بیشترین تعداد دفعات استفاده را دارد.
- در مورد زمان استفاده از I/O تصمیم می گیرد.
- میزان تخصیص CPU به یک کار را مشخص می کند.

دلایل تغییر سیستم عامل در طول زمان:

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

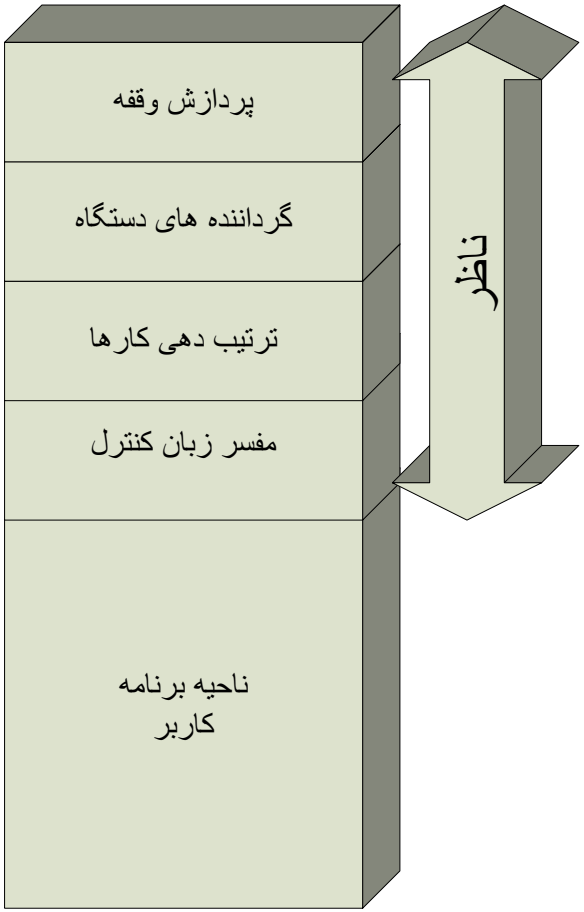
تکامل تدریجی سیستم عامل (پردازش ردیفی) :

- سیستم عاملی وجود نداشت.
- برنامه ها به زبان ماشین بر روی کارت نوشته می شد و به دستگاه ورودی (نوار خوان) بار می شد.
- کاربر با سخت افزار در ارتباط بود.
- دارای یک میز فرمان بود.
- در صورت ایجاد خطا کاربر ثبات ها و حافظه اصلی را بررسی می کرد.
- دومساله اصلی :
- **زمانبندی** : هر کاربر باید از برگه های نوبت گیری استفاده کند
- **زمان نصب** : هر برنامه شامل بار کردن مترجم و کد منبع به حافظه، ذخیره سازی برنامه ترجمه شده بود

تکامل تدریجی سیستم عامل:(پردازش 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.

وضعیت حافظه برای ناظر ماندگار:



وضعیت حافظه
برای یک 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 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)

حافظلت 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 .

Multiprogramming Example

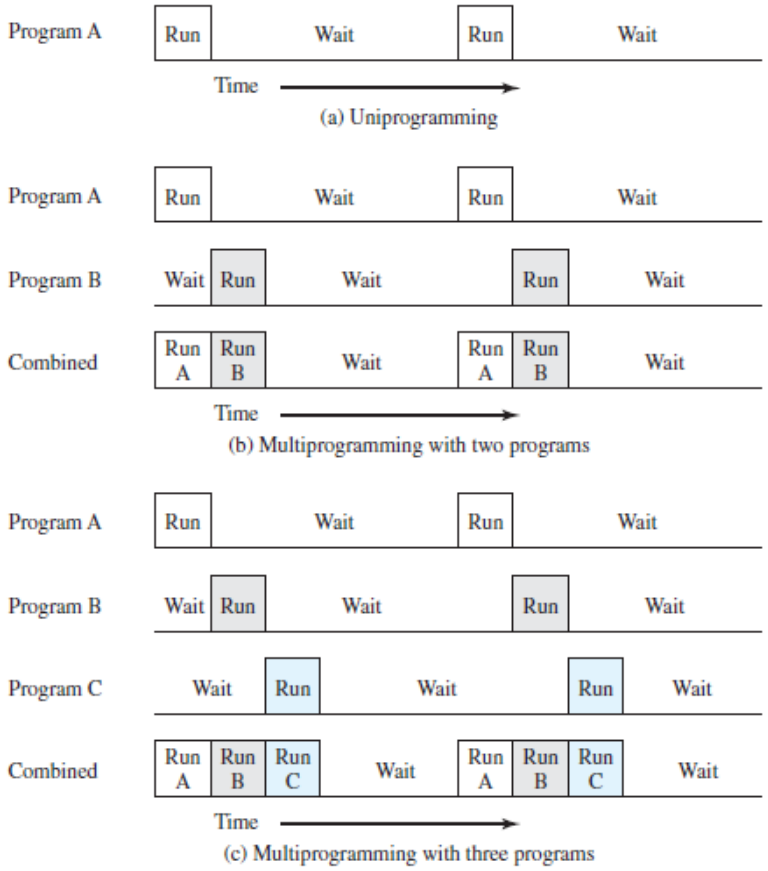


Figure 2.5 **Multiprogramming Example**

سیستم عامل اشتراک زمانی:

- در بعضی مواقع مثل پردازش تراکنش حالت محاوره ای ضروری است. لذا سیستم های اشتراک زمانی بوجود آمدند.

- چند برنامگی امکان رسیدگی به کار های محاوره ای را میدهد. وقت پردازنده بین کارها تقسیم میشود.ار این روش در صورت وجودn کاربر به هر یک از کاربران1/n وقت پردازنده میرسد .

مقایسه چند برنامه ای و اشتراک زمانی:

- **هدف**
  - از چند برنامه ای دسته ای : استفاده حداکثر از پردازنده
  - از اشتراک زمانی : حداقل زمان پاسخ
- **منبع دستورات به سیستم عامل**
  - در چند برنامه ای: دستورات زبان کنترل کار
  - در اشتراک زمانی : فرمان هایی که از پایانه وارد می شوند.

CTSS(Compatible Time Sharing System)

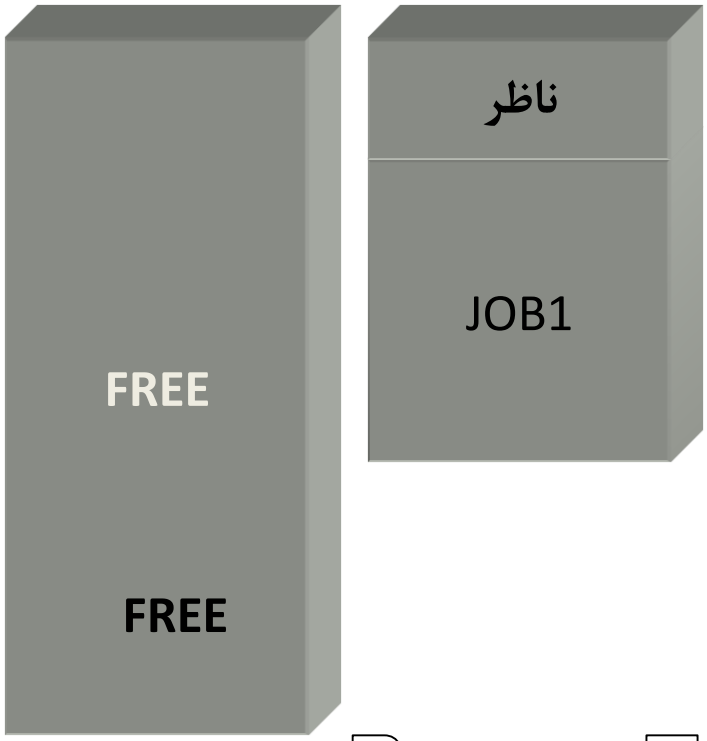
- این سیستم روی ماشینی با ۳۲۰۰۰ کلمه حافظه ۳۲ بیتی اجرا می شد. ۵۰۰۰ کلمه برای ناظر ماندگار و ۲۷۰۰۰ کلمه برای برنامه کاربر و داده هایش
- یک برنامه همیشه طوری بار می شود که از آدرس ۵۰۰۰ شروع شود بنابراین کار ناظر و هم مدیریت حافظه آسان تر می شد.

- 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 :

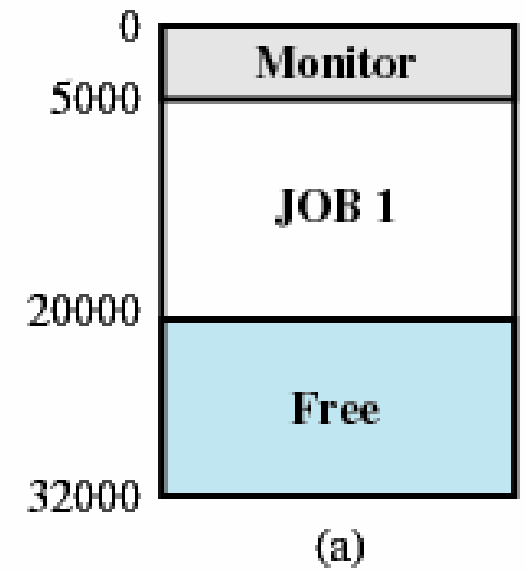
- مثال:اگر تعداد کلمات مورد نیاز هر کار بصورت زیر باشد نحوه انجام عملیات به صورت زیر است.
  - Job1: 15000 Job2: 20000
  - Job3: 5000 Job4: 10000
- ترتیب انجام کار ها :
  -

مثالی برای درک عملکرد CTSS:



- ۵۰۰۰ کلمه اول را ناظر اشغال میکند و

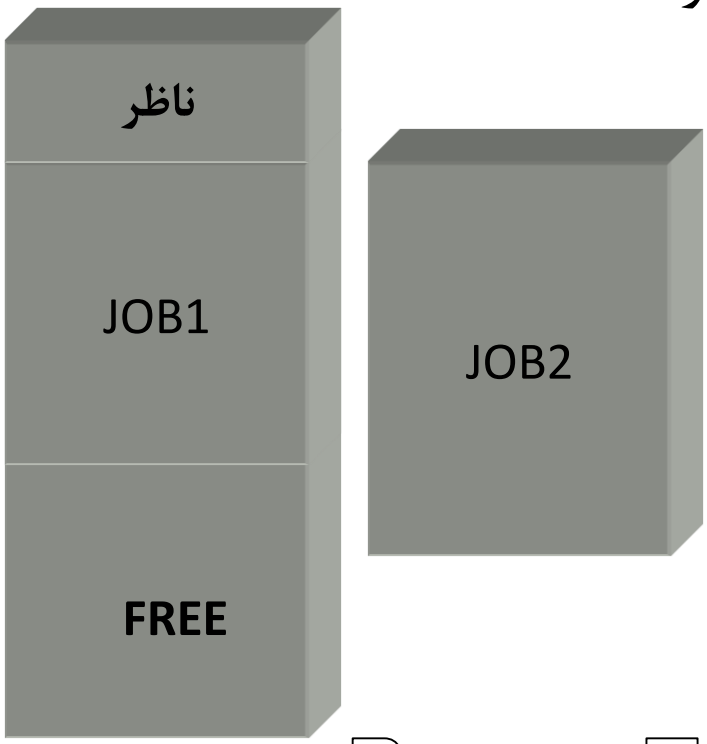
از محل ۵۰۰۰ Job1 بار میشود.

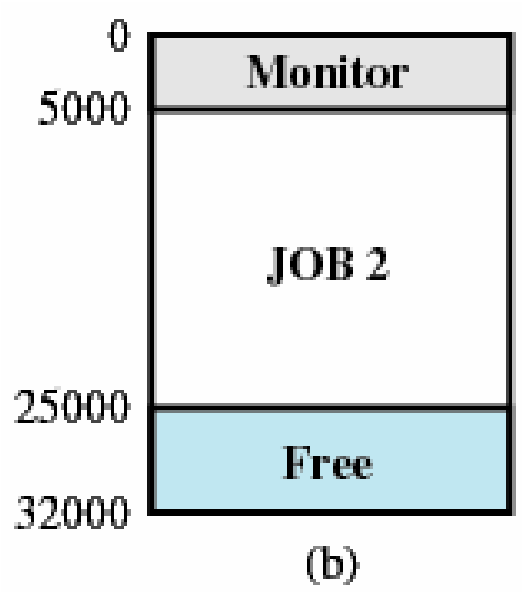


Press Enter

مثالی برای درک عملکرد CTSS:

- چون Job2 از Job1بزرگتراست،
  کل Job1 بر روی دیسک ذخیره میشود
  و Job2 بار میشود.

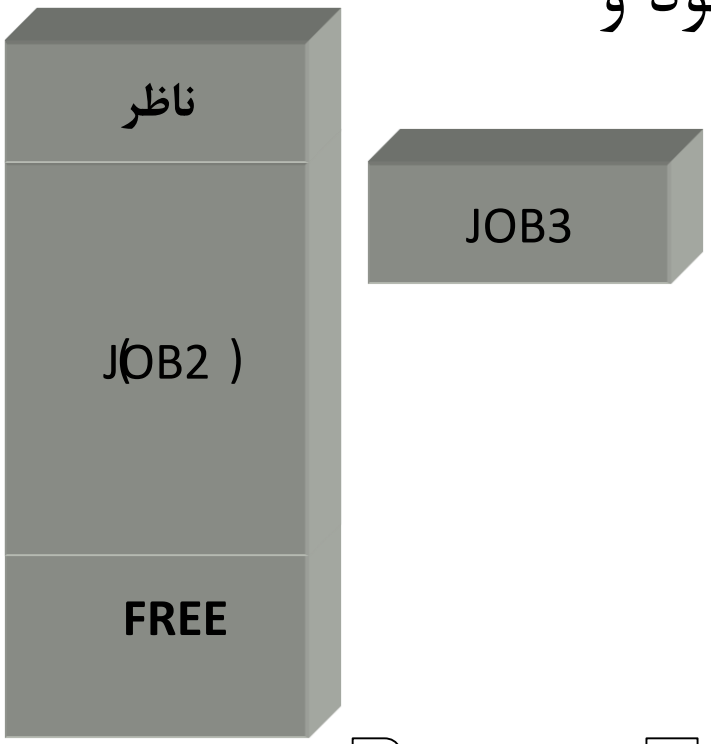


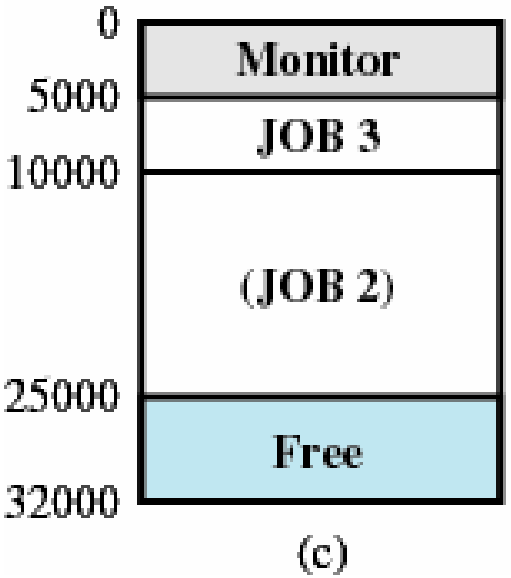


Press Enter

مثالی برای درک عملکرد CTSS :

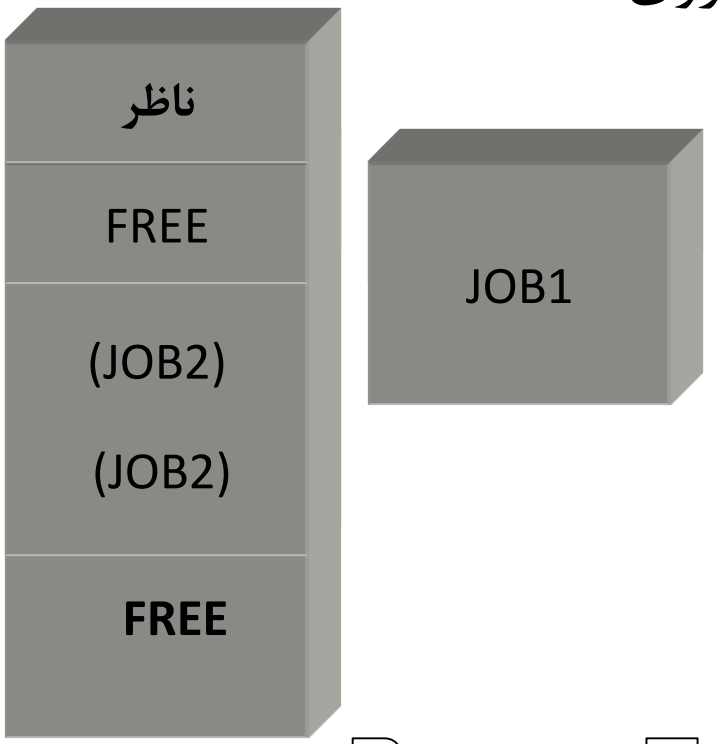
- چون Job3 از Job2 کوچکتر است مقداری از Job2 بر روی دیسک ذخیره میشود و Job3 از محل ۵۰۰۰بار میشود.

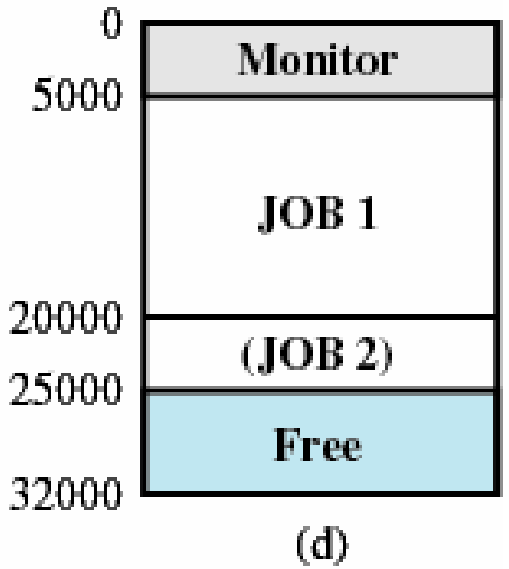




مثالی برای درک عملکرد CTSS :

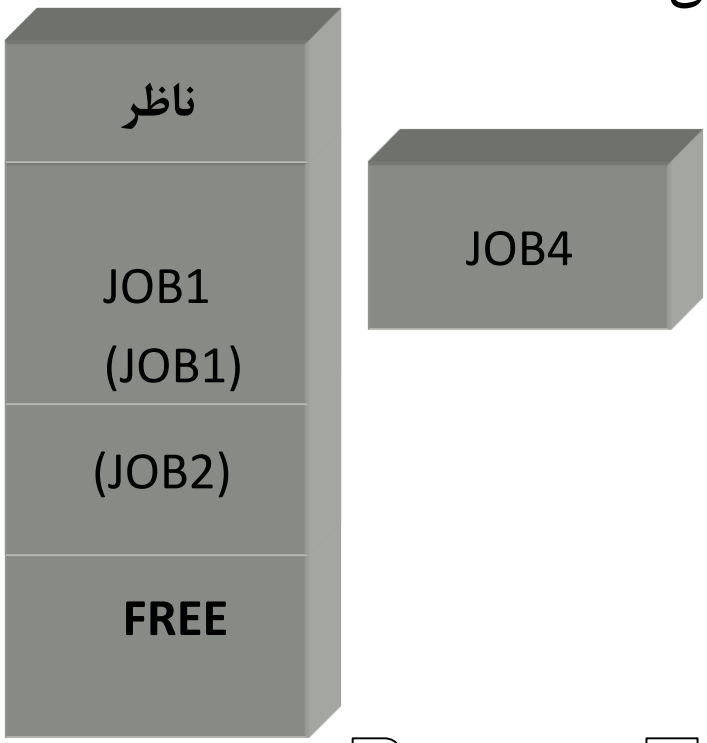
- Job1 بزرگتر از Job3 و کوچکتر از Job2 است، کل Job3 و مقداری از Job2 روی دیسک ذخیره میشود و Job1 از محل ۵۰۰۰ بار میشود.

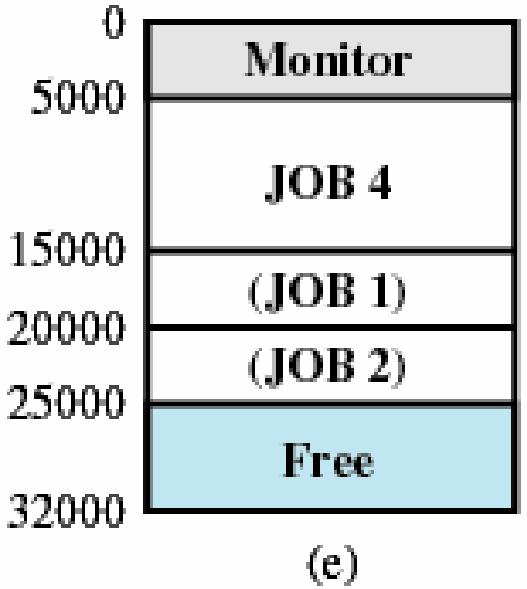




مثالی برای درک عملکرد CTSS :

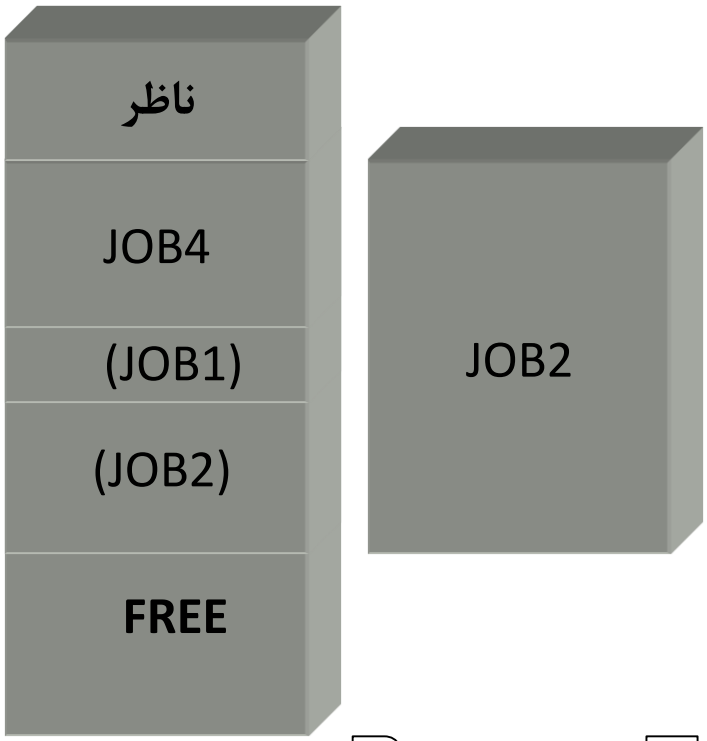
- چون Job4 کوچکتر از Job1، مقداری از Job1 روی دیسک ذخیره میشود و کل Job4 از محل ۵۰۰۰ به بعد بار می شود

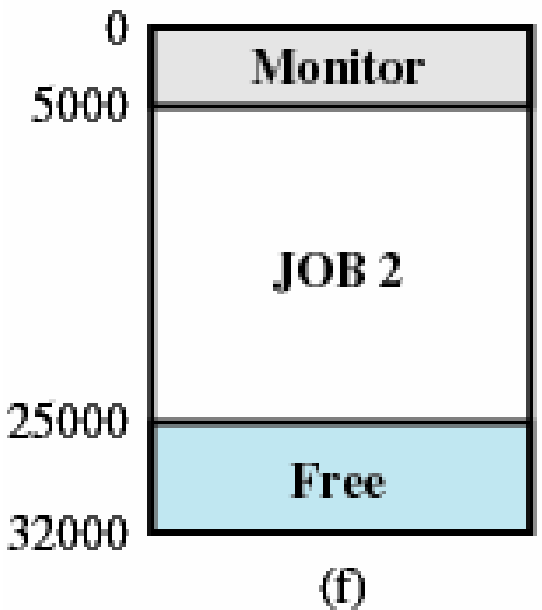




مثالی برای درک عملکرد CTSS:

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





پایان جلسه چهارم