دانشگاه صنعتی شاهرود
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

سیستم‌های عامل

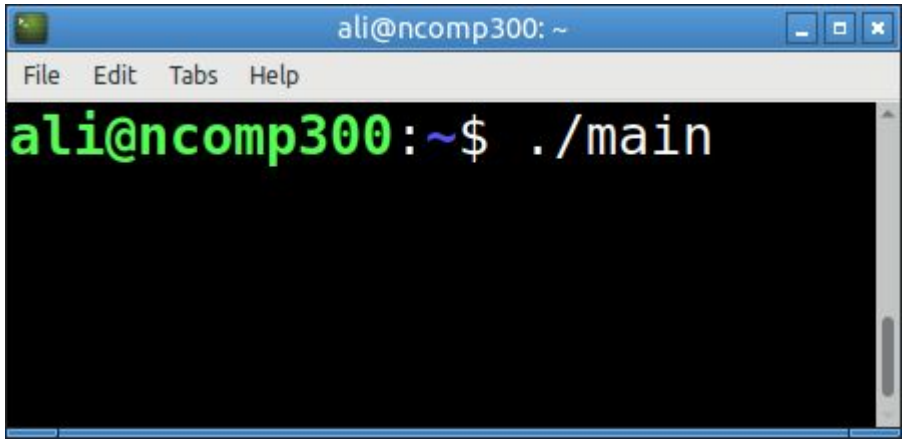
معرفی فرآیندها

استاد: علیرضا تجری

معرفی برنامه program

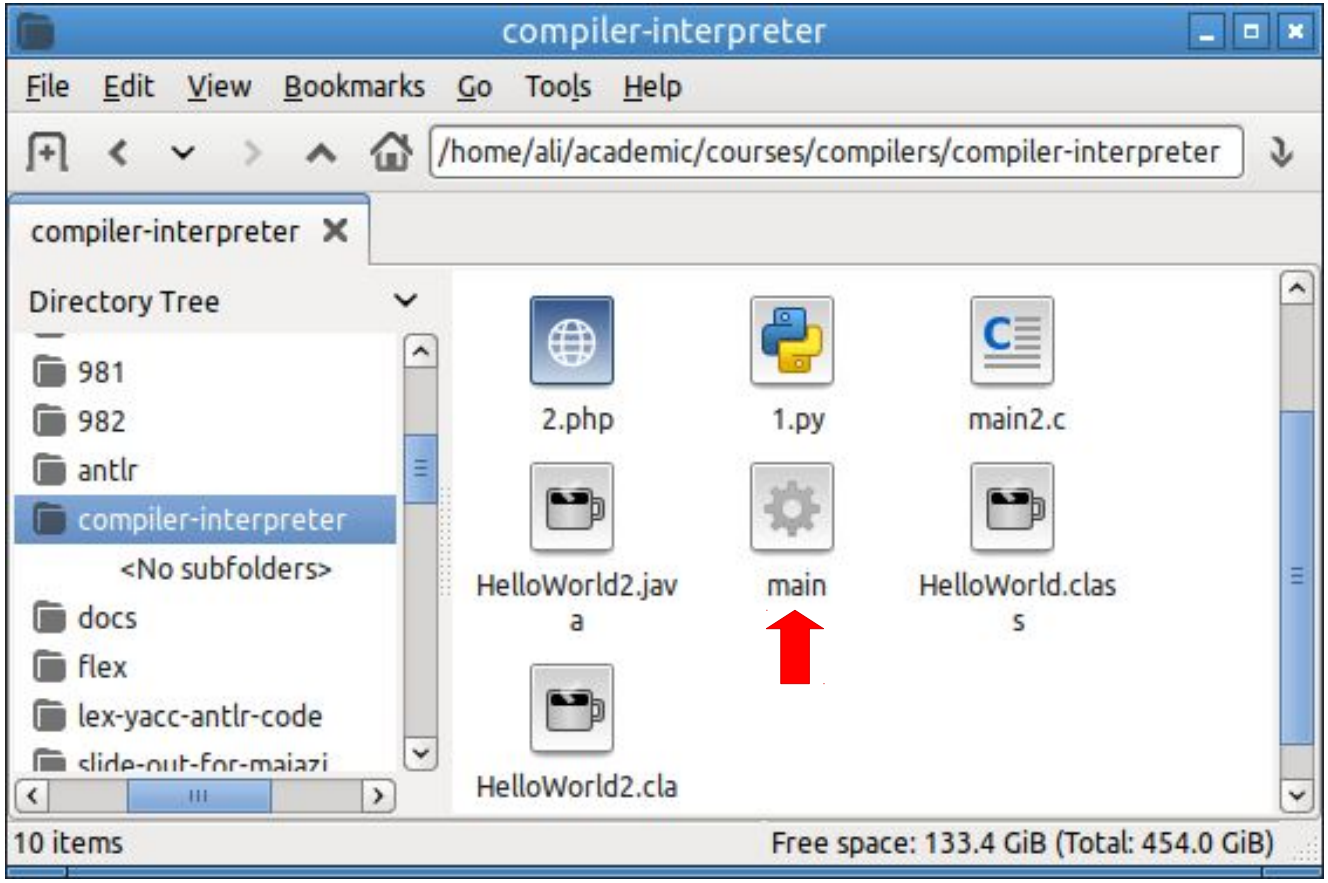
notepad.exe

- فایلی که می‌توان آن را اجرا کرد.
  - بدون نیاز به مفسر
  - با دابل کلیک یا اجرا در مفسر فرمان
    - با دابل کلیک بر روی shortcut
  - مثال
    - فایل‌های با پسوند exe
    - فایل کامپایل شده یک فایل c



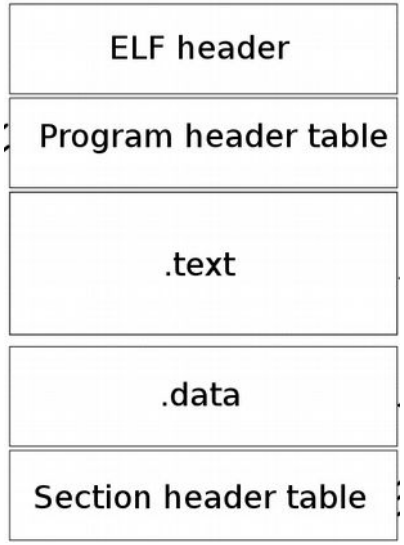
معرفی برنامه program

- برنامه‌ها در کجا قرار دارند؟
  - در دیسک سخت
    - در فایل سیستم
      - در پوشه‌ها
- برنامه
  - یک فایل
  - قابل اجرا شدن



معرفی برنامه program

- فایل‌های اجرایی دارای ساختارهای استاندارد هستند.
  - همانطورکه فایل‌های عکس یا ویدئو یا ورد یا html دارای ساختار هستند.

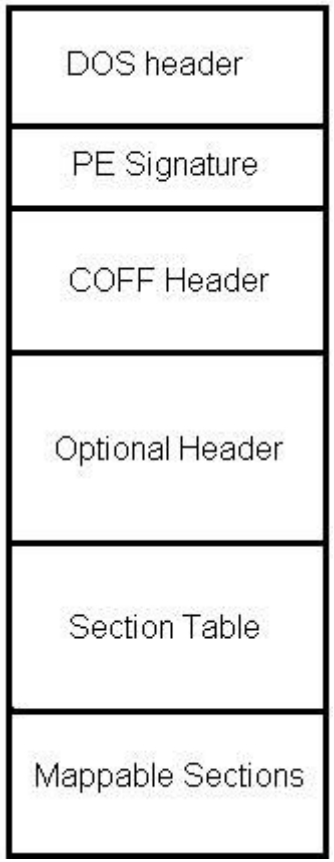


- مثال: ساختار فایل‌های اجرایی در لینوکس
  - در هر بخش تعدادی بایت داده قرار دارد.

```
00000000 7F 45 4C 46 02 01 01 00 00 00 00 00 00 00 00 00 .ELF............
0000000f 00 03 00 3E 00 01 00 00 00 40 05 00 00 00 00 00 ...>.....@......
0000001e 00 00 40 00 00 00 00 00 00 00 30 19 00 00 00 00 ..@.......0.....
0000002d 00 00 00 00 00 00 00 40 00 38 00 09 00 40 00 00 .......@.8...@..
0000003c 1D 00 1C 00 06 00 00 00 04 00 00 00 40 00 00 00 ............@...
```

معرفی برنامه program

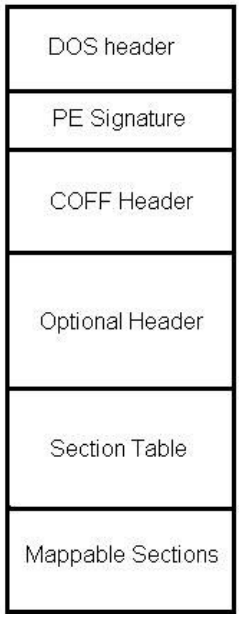
- فایل‌های اجرایی دارای ساختارهای استاندارد هستند.
  - این ساختار وابسته به سیستم‌عامل است.

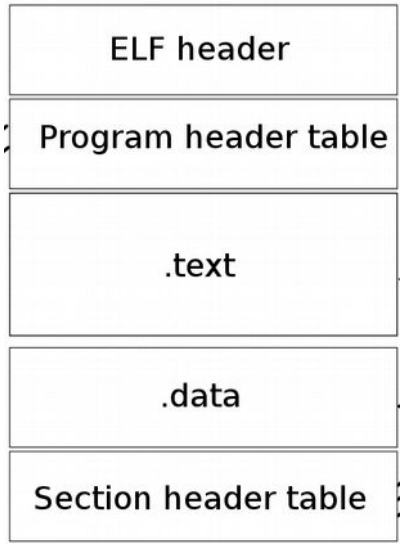



```
00000000 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 MZ..............
0000000f 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 .........@.....
0000001e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...............
0000002d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...............
0000003c F0 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 ............!..
0000004b 4C CD 21 54 68 69 73 20 70 72 6F 67 72 61 6D L.!This program
0000005a 20 63 61 6E 6E 6F 74 20 62 65 20 72 75 6E 20  cannot be run
00000069 69 6E 20 44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A in DOS mode....
00000078 24 00 00 00 00 00 00 00 4B E3 16 1D 0F 82 78 $.......K.....x
```

معرفی برنامه program

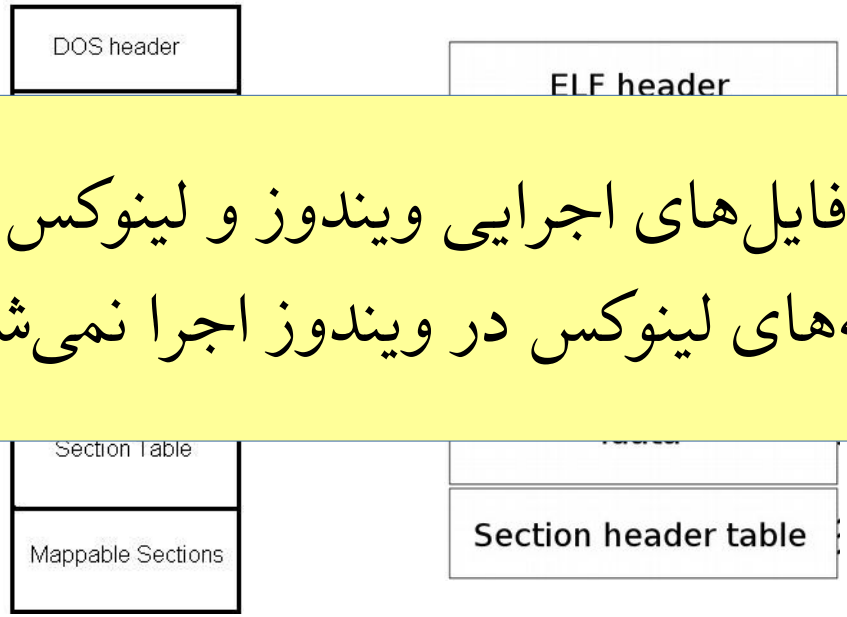
- فایل‌های اجرایی دارای ساختارهای استاندارد هستند.
  - این ساختار وابسته به سیستم‌عامل است.
  - به همین دلیل برنامه‌های اجرایی ویندوز در لینوکس اجرا نمی‌شود و بالعکس.





معرفی برنامه program

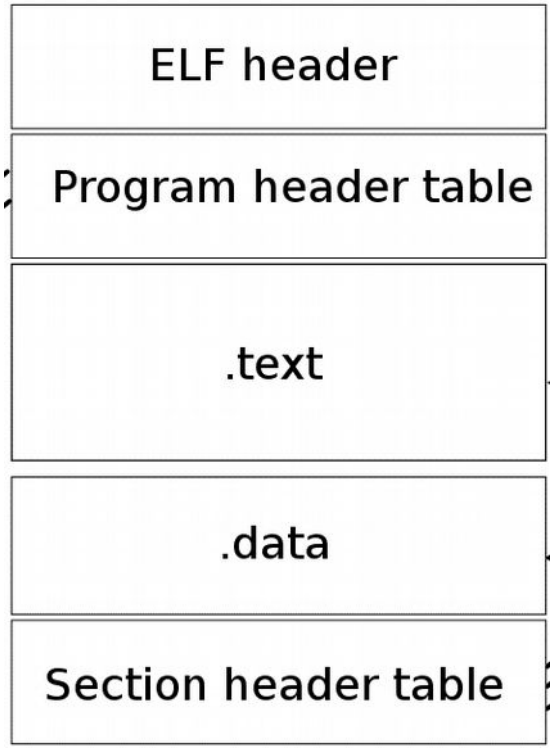
- فایل‌های اجرایی دارای ساختارهای استاندارد هستند.
  - این ساختار وابسته به سیستم‌عامل است.
  - به همین دلیل برنامه‌های اجرایی ویندوز در لینوکس اجرا نمی‌شود و بالعکس.



حتی اگر فرمت فایل‌های اجرایی ویندوز و لینوکس یکسان بود، باز هم برنامه‌های لینوکس در ویندوز اجرا نمی‌شد. چرا؟

معرفی برنامه program

- فایل برنامه دارای بخش‌های مختلفی است.



- تعدادی سرآیند. مانند:
  - معماری پردازنده‌ای که می‌تواند این برنامه را اجرا کند
  - کتابخانه‌های مورد استفاده
- بخش‌های مربوط به کد و داده
  - بخش کد: این بخش با text. شناخته می‌شود.
    - کد به زبان ماشین که پردازنده می‌تواند آن را اجرا کند.
  - بخش داده: این بخش با data. شناخته می‌شود.
    - داده‌های global برنامه. متغیرهایی که خارج از main تعریف شده‌اند.

وقتی یک برنامه را اجرا می‌کنیم، چه اتفاقی می‌افتد؟

- سیستم‌عامل بررسی می‌کند که آیا می‌تواند فایل را اجرا کند یا خیر
  - فرمت فایل درست است؟ می‌توان روی این پردازنده اجر شود؟ و ...
- سیستم‌عامل یک فضای خالی در حافظه اصلی برای این برنامه در نظر می‌گیرد.
- بخش **text**. و **data**. را از فایل برنامه به فضای برنامه در حافظه اصلی کپی می‌کند.
- نرم‌افزاری به نام بارکننده، کتابخانه‌های مورد نیاز را در حافظه اصلی قرار می‌دهد.
- برنامه آماده اجرا است.

وقتی یک برنامه را اجرا می‌کنیم، چه اتفاقی می‌افتد؟

- سیستم‌عامل بررسی می‌کند که آیا می‌تواند فایل را اجرا کند یا خیر
- س
- ب
- ن
- برنامه اماده اجرا است.

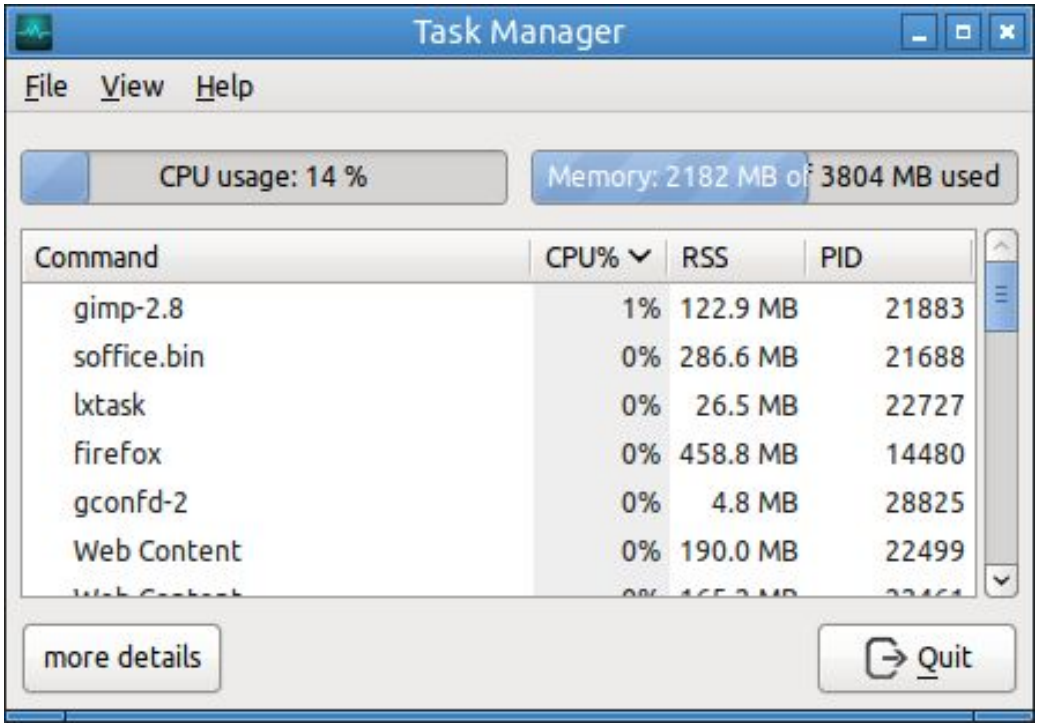
به برنامه‌ای که در حافظه قرار گرفته است،

فرآیند (Process)

می‌گوییم.

فرآیند چیست؟

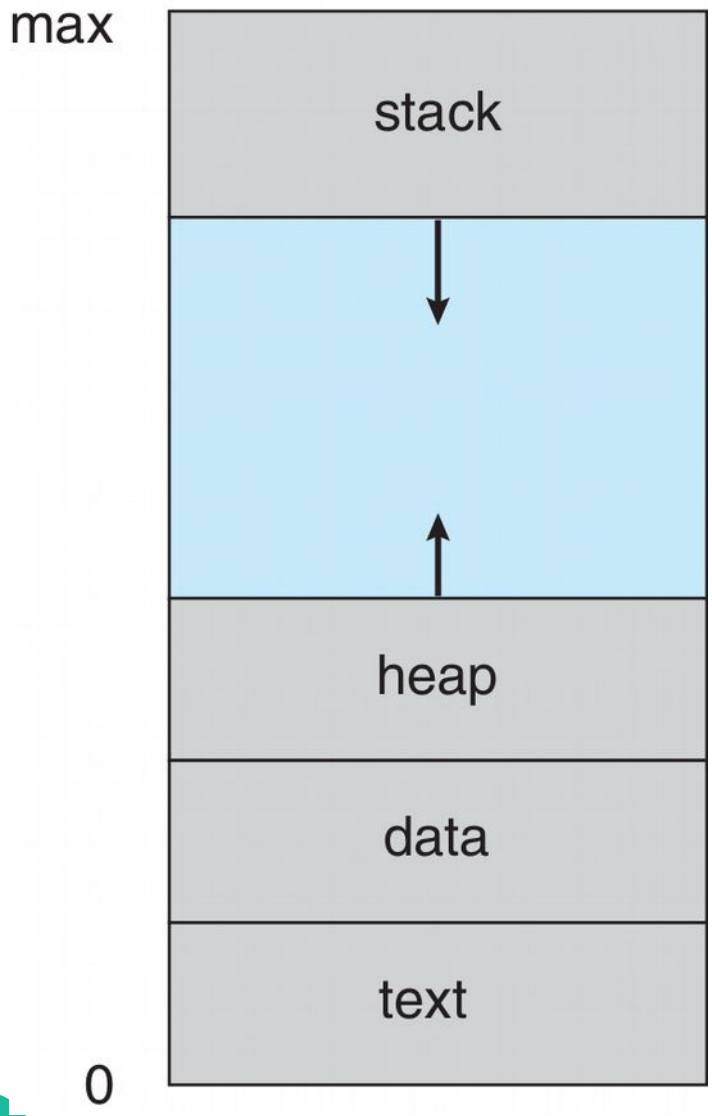
- برنامه‌ای است که در حافظه قرار گرفته است.
- برنامه‌ای که دستورات آن اجرا می‌شود.
- فرآیندها در کجا قرار دارند؟
  - حافظه اصلی
- برنامه‌ها در کجا قرار دارند؟
  - فایل سیستم
- می‌توان فرآیندها را دید!
  - با استفاده از Task Manager

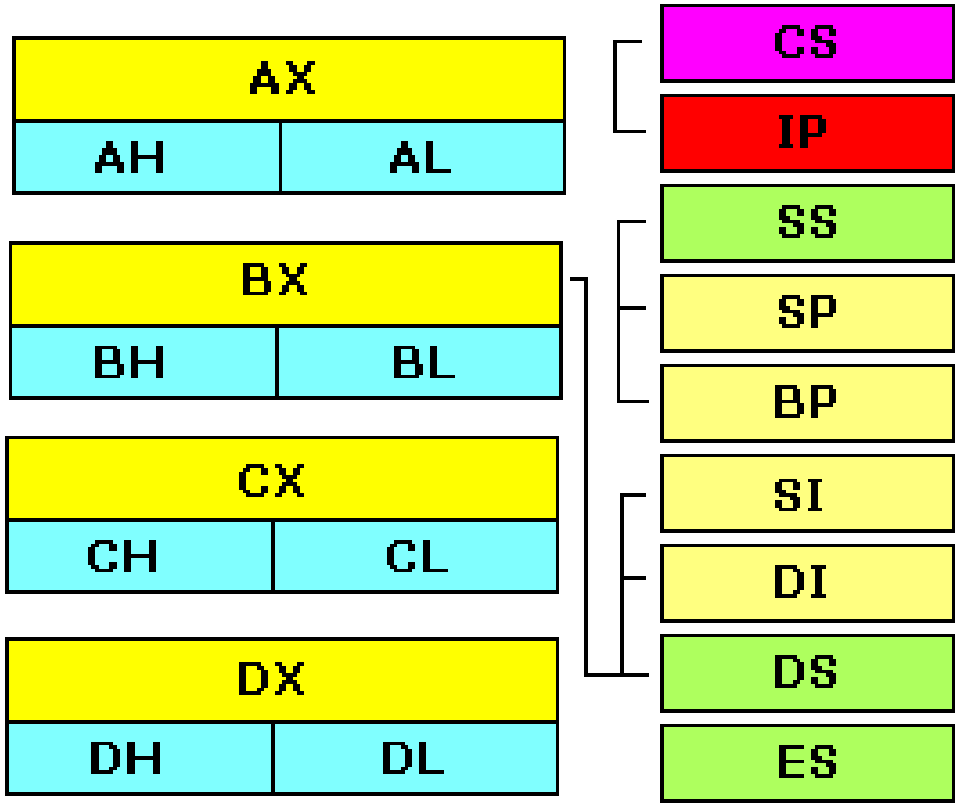


فرآیند شامل چه چیزهای است؟

- کد: دستورالعمل‌هایی به زبان ماشین که پردازنده آن‌ها را به صورت پشت‌سرهم اجرا می‌کند (**text**).
- داده: داده‌های **global** ای که بخش کد از آن‌ها برای ذخیره کردن متغیرها استفاده می‌کند (**data**).
- حافظه پویا: بخش از حافظه که به صورت پویا (با استفاده از **new**) اختصاص داده شده است (**heap**).
- پشته: بخشی از حافظه که آدرس‌های برگشت از توابع و پارامترهای محلی تابع در آن ذخیره می‌شود (**Stack**).
- رجیسترهای پردازنده: رجیسترهایی که در دستورالعمل‌های پردازنده از آن‌ها استفاده می‌شود.
  - رجیستر PC (در برخی از پردازنده‌های، نام این رجیستر IP است.)
  - رجیسترهای همه منظوره

بخش‌های فرآیند در حافظه اصلی و پردازنده





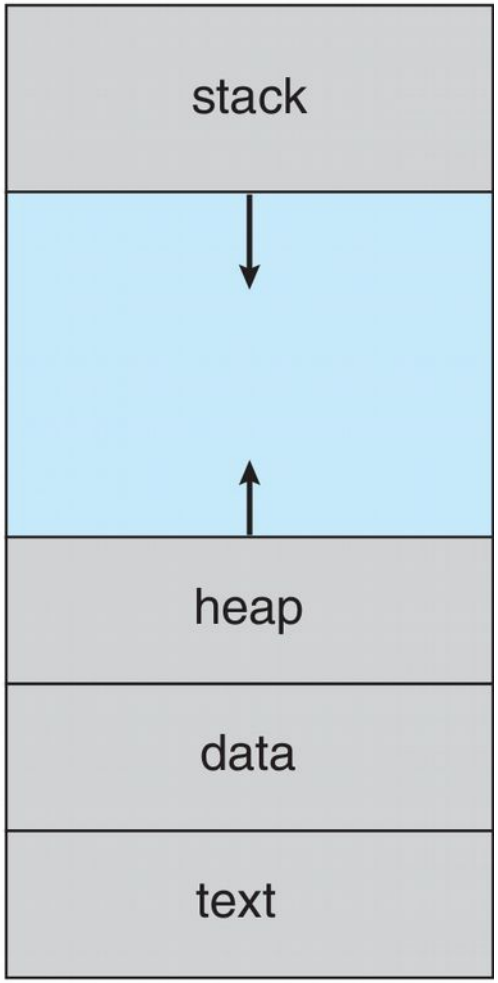
از اسلایدهای کتاب سیلبرشاتز

فرآیند چه تفاوتی با برنامه دارد؟

- برنامه ماهیت غیرفعال **passive** دارد.
  - اگر بعد از چند روز به برنامه نگاه کنیم، هیچ تغییری نکرده است.
  - متغیرهای درون بخش data برنامه تغییری نمی‌کنند.
- فرآیند ماهیت فعال **active** دارد.
  - برنامه در هر یک ثانیه میلیون‌ها تغییر می‌کند.
  - متغیرهای درون بخش data فرآیند تغییر می‌کنند.

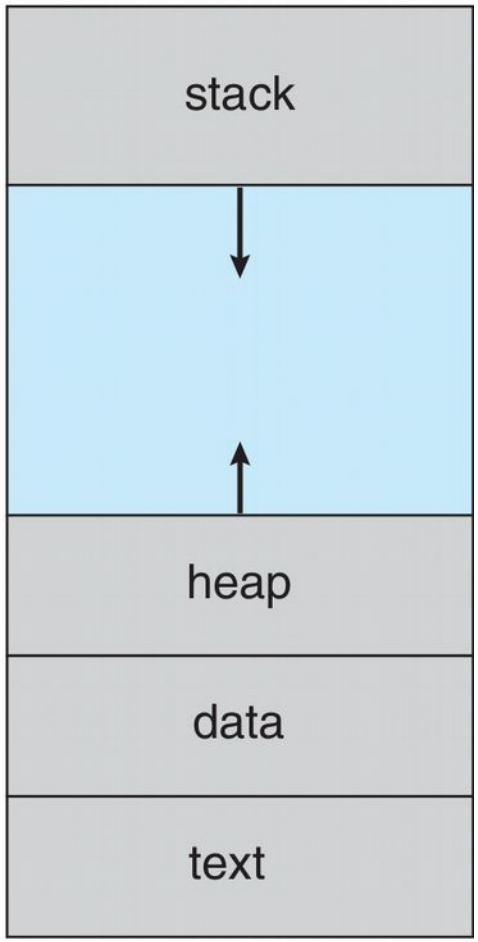
می‌توان از روی یک برنامه، چند فرآیند ایجاد کرد.

پس از اینکه فرآیند به وجود آمد، چه اتفاقی می‌افتد؟



- در ابتدا،
  - داده‌ای در بخش stack و heap وجود ندارد.
  - مقدار اولیه متغیرهای گلوبال در data وجود دارد.
  - کد برنامه در text قرار دارد.
- پس از آن برنامه می‌تواند اجرا شود.
  - توسط؟ پردازنده
  - از کجا؟ از یک مکان خاص در حافظه (تابع main)
  - برنامه الان به چه زبانی است؟ زبان ماشین (زبان پردازنده)
- دستورالعمل‌ها به صورت پشت‌سرهم اجرا می‌شوند.

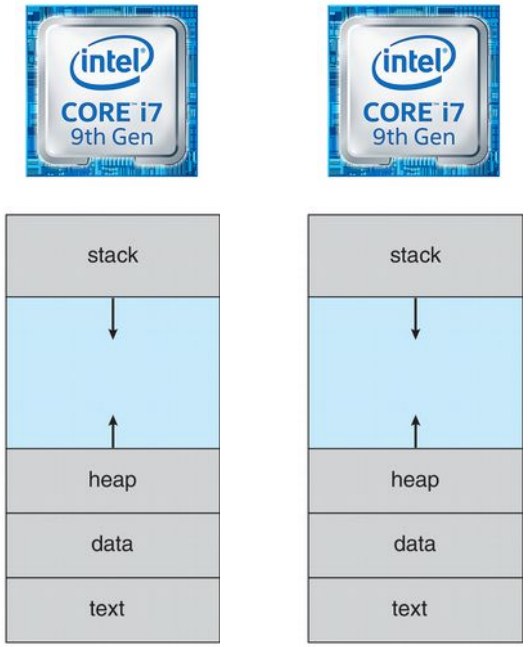
پس از اینکه فرآیند به وجود آمد، چه اتفاقی می‌افتد؟



- دستورالعمل‌ها به صورت پشت‌سرهم اجرا می‌شوند.
  - در حین اجرا،
    - مقدار رجیستر PC تغییر می‌کند.
    - مقدار رجیسترهای دیگر پردازنده تغییر می‌کند.
      - بستگی به دستوراتی دارد که اجرا می‌شود.
    - اگر داده‌ای در متغیرهای گلوبال نوشته شود،
      - حافظه متغیرهای گلوبال موجود در بخش data تغییر می‌کند.
    - اگر تابعی صدا زده شود، بخش stack تغییر می‌کند.
    - اگر شیءایی new یا delete شود، بخش heap تغییر می‌کند.
  - پس از صدا زدن فراخوانی سیستمی exit، اجرا به پایان می‌رسد.

اگر از یک برنامه، دو فرآیند به وجود بیاید، چگونه اجرا می‌شوند؟

- آیا اینکار امکان دارد؟
  - بله، برنامه را دو بار اجرا می‌کنیم. (مثلاً دو notepad باز می‌کنیم.)
- در ابتدا، حافظه هر دو فرآیند مثل هم است.
- با کار کردن با یک فرآیند،
  - حافظه آن تغییر می‌کند.
    - بخش کد (text) ثابت باقی می‌ماند.
  - رجیسترهای پردازنده تغییر می‌کنند.
    - شماره دستوری که اجرا می‌شود و ...



آیا به محض قرار گرفتن برنامه در حافظه، اجرا می‌شود؟

- وقتی که برنامه در حافظه قرار گرفت (فرآیند ایجاد شد)،
  - فرآیندهای دیگری هم در سیستم وجود دارد.
    - چندبرنامگی Multi-Programming
  - پردازنده در حال اجرای دستورالعمل‌های یک فرآیند دیگر است.
    - سیستم‌عامل
- چه زمانی فرآیند جدید اجرا می‌شود؟
- چه زمانی فرآیندهای دیگر اجرا می‌شوند؟

سیستم‌عامل، اجرای فرآیندها را با استفاده از **تعدادی صف**، مدیریت می‌کند.

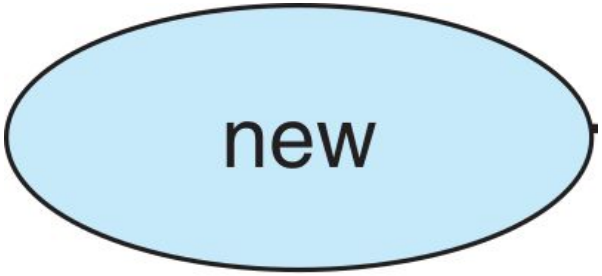
در هر لحظه،
پردازنده،
فقط یک دستورالعمل
از یک فرآیند
را اجرا می‌کند.

چگونه اجرای فرآیندها مدیریت می‌شود؟

- سیستم‌عامل، اجرای فرآیندها را با استفاده از **تعدادی صف**، مدیریت می‌کند.
  - به عنوان مثال، صف آماده
    - یک صف که در آن، فرآیندهایی که آماده اجرا هستند، قرار می‌گیرند.
    - فرآیندهای جدید به این صف اضافه می‌شوند.
    - سیستم‌عامل، هر بار یک فرآیند از این صف انتخاب می‌کند
      - طی عملیاتی، پردازنده، آن فرآیند را اجرا می‌کند.
      - تا زمانی که
        - اتفاقی در سیستم رخ دهد
        - یا نتوان آن فرآیند را اجرا کرد.

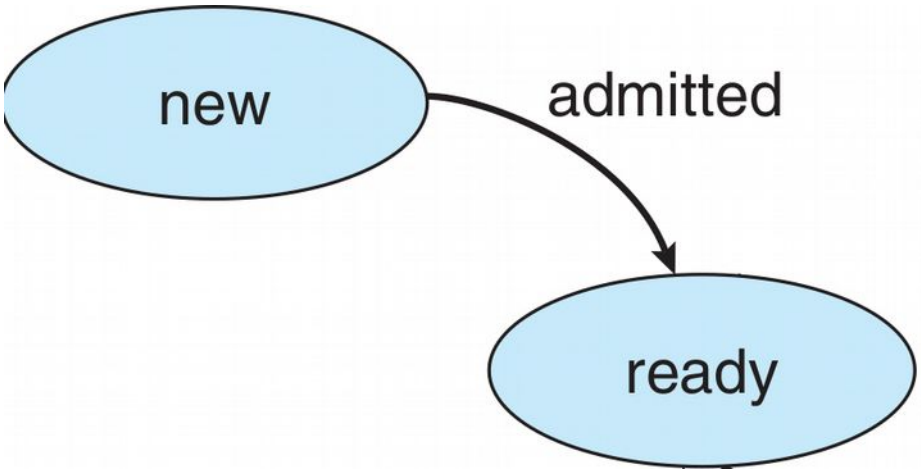
وضعیت فرآیند Process State

- هر فرآیند در حین اجرا، دارای وضعیت‌های متفاوتی است.
- وضعیت جدید (**New**): زمانی که فرآیند ایجاد شده است.
  - حافظه به آن اختصاص داده شده است.
  - هنوز هیچ دستوری از این فرآیند اجرا نشده است.



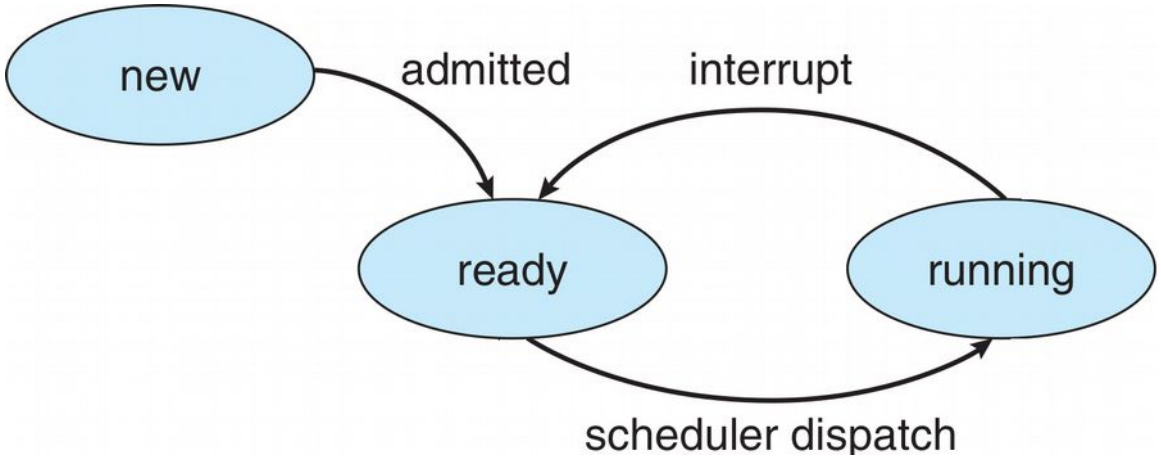
وضعیت فرآیند Process State

- وضعیت آماده (**Ready**): آماده اجرا شدن توسط پردازنده
  - اجرای فرآیند، مورد قبول سیستم‌عامل قرار گرفت.
  - هنوز هم هیچ دستورالعملی اجرا نشده است.



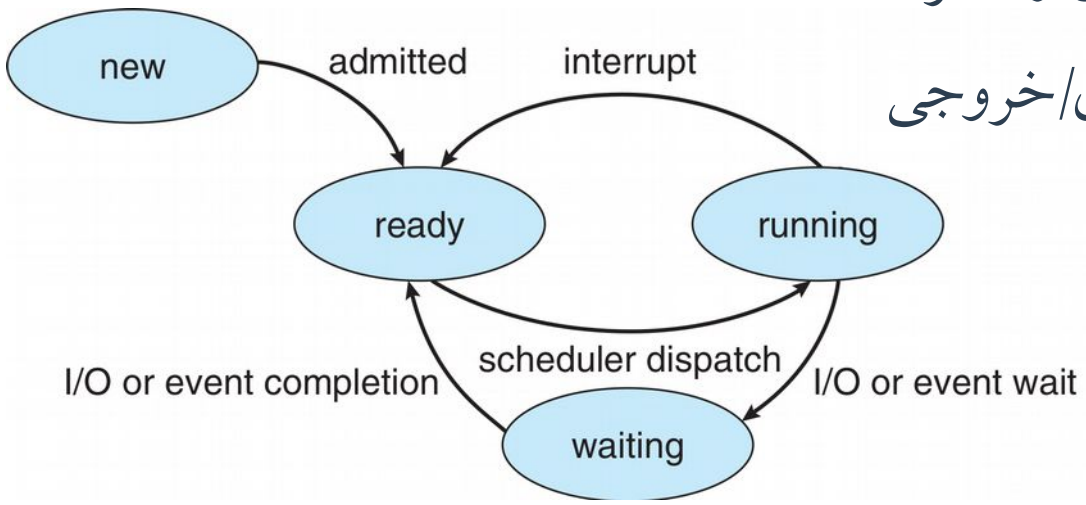
وضعیت فرآیند Process State

- وضعیت در حال اجرا (**Running**): در حالی اجرا شدن دستورالعمل‌ها توسط پردازنده
  - سیستم‌عامل تصمیم گرفت که این فرآیند بر روی پردازنده اجرا شود.
  - اگر وقفه‌ای رخ بدهد، باید کد سیستم‌عامل اجرا شود (روتین سرویس وقفه)، بنابراین، فرآیند به وضعیت آماده برمی‌گردد.



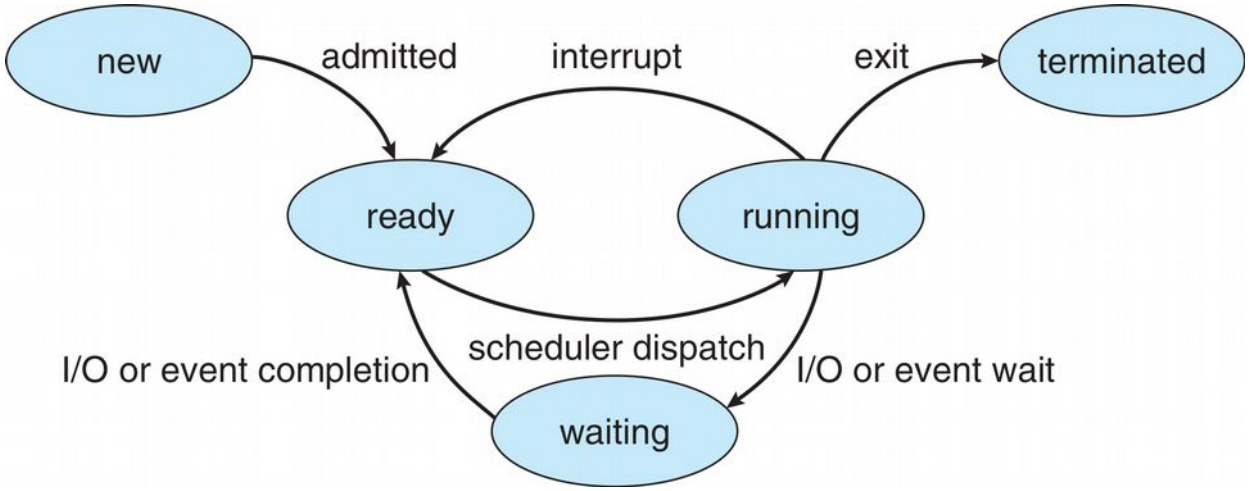
وضعیت فرآیند Process State

- وضعیت انتظار (**Waiting**): منتظر یک رخداد یا انجام عملیات ورودی/خروجی
  - فرآیند درخواست یک عملیات ورودی/خروجی دارد.
    - زمانبر است / توسط سخت‌افزار دیگری اجرا می‌شود.
    - پردازنده باید فرآیند دیگری را اجرا کند.
    - پس انجام عملیات ورودی/خروجی وضعیت فرآیند، Ready می‌شود.

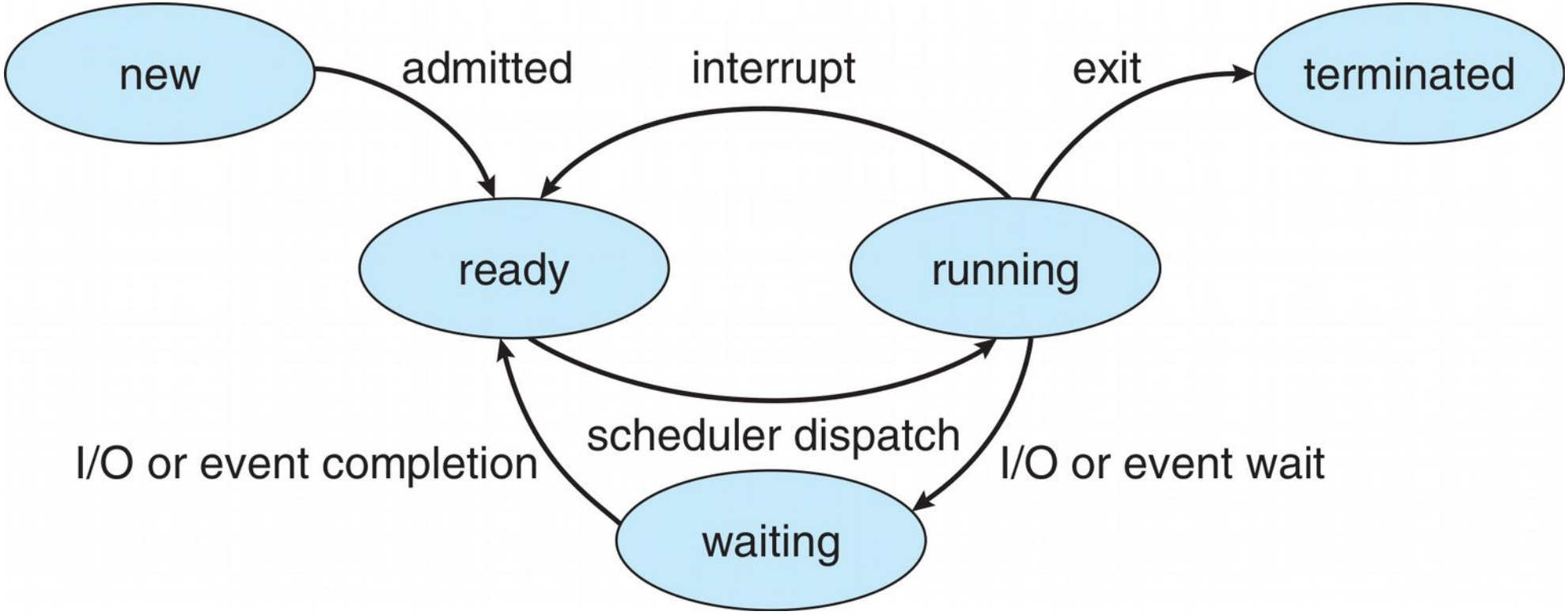


وضعیت فرآیند Process State

- وضعیت تمام شده (**Terminated**): اجرای فرآیند به اتمام رسیده
  - فرآیند فراخوانی سیستمی exit را صدا زده
  - در اجرای فرآیند مشکلی پیش آمده که نمی‌توان اجرا را ادامه داد.



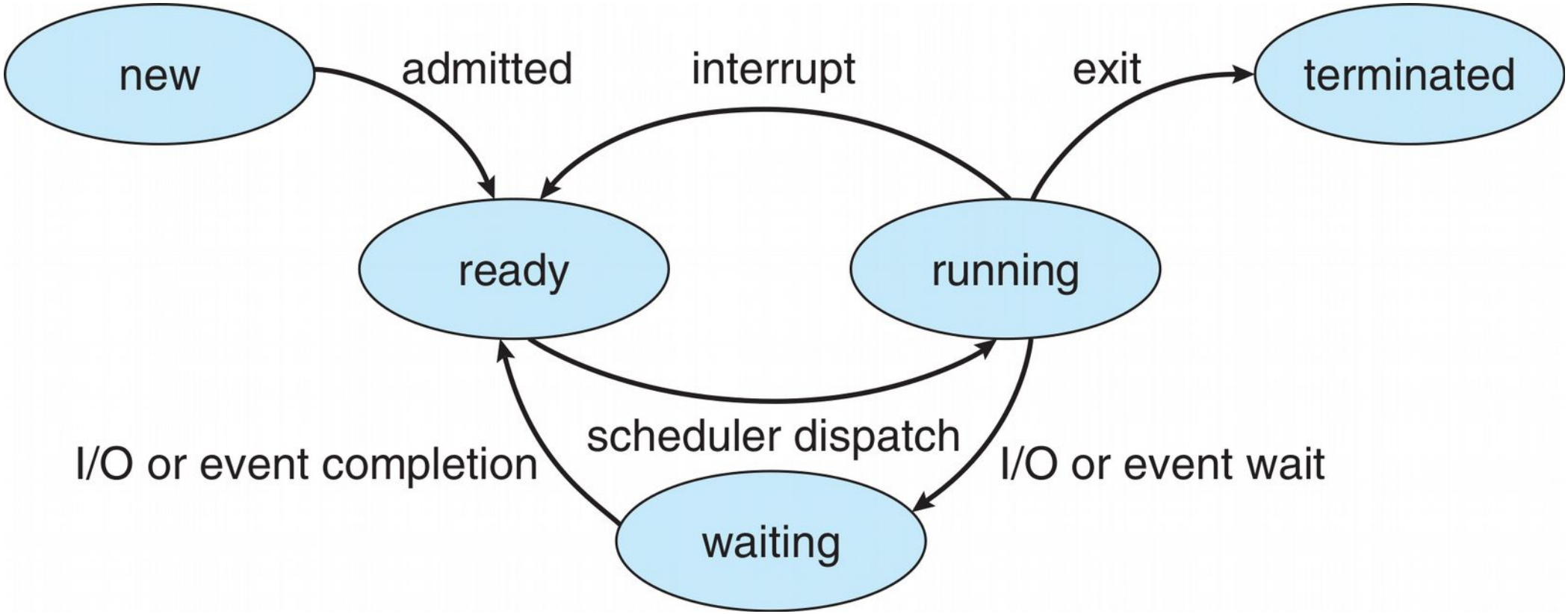
وضعیت فرآیند Process State



سیستم‌عامل‌های مختلف، ممکن است نمودارهای متفاوتی داشته باشند.

از اسلایدهای کتاب سیلبرشاتز

وقتی چند فرآیند داریم، هر یک در چه state ای است؟



بلوک کنترل فرآیند Process Control Block

- یک ساختار داده که سیستم‌عامل برای هر فرآیند نگهداری می‌کند.
  - در حافظه اصلی
- چه کاربردی دارد؟
  - همه اطلاعات مربوط به فرآیند در آن موجود است.
    - وضعیت فرآیند چیست؟
    - مقدار رجیسترهای پردازنده چیست؟
    - چه کاربری این فرآیند را ایجاد کرده است؟
    - این فرآیند چه فایل‌هایی را باز کرده است؟
    - و ...

بلوک کنترل فرآیند Process Control Block

- چه فیلدهایی دارد؟
  - حالت فرآیند Process State
  - مقدار رجیستر شمارنده برنامه Program Counter
  - مقدار رجیسترهای دیگر پردازنده CPU Registers

وضعیت پردازنده
Process State

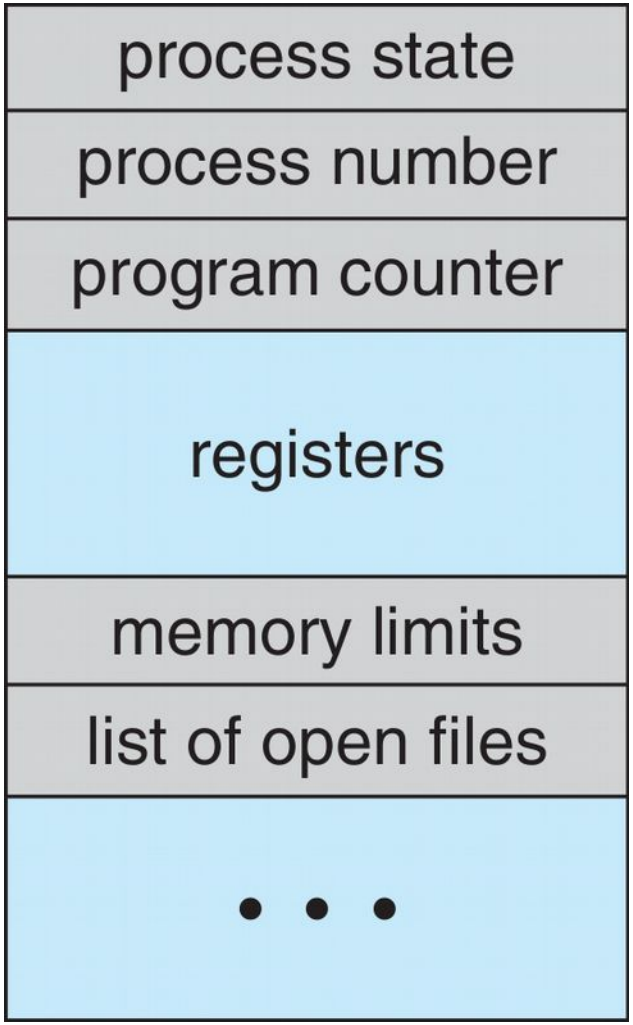
  - اطلاعات زمان‌بندی پردازنده CPU Scheduling Information
    - اولویت‌های فرآیند و ...

بلوک کنترل فرآیند Process Control Block

- چه فیلدهایی دارد؟

- اطلاعات مدیریت حافظه Memory Management Information
  - چه بخشی از حافظه به فرآیند اختصاص داده شده است؟
- اطلاعات حسابرسی Accounting Information
  - شناسه فرآیند
  - چه کاربری در چه زمانی فرآیند را اجرا کرده و ...
- اطلاعات وضعیت ورودی/خروجی IO/ State Information
  - فایل‌های باز شده توسط فرآیند و ...

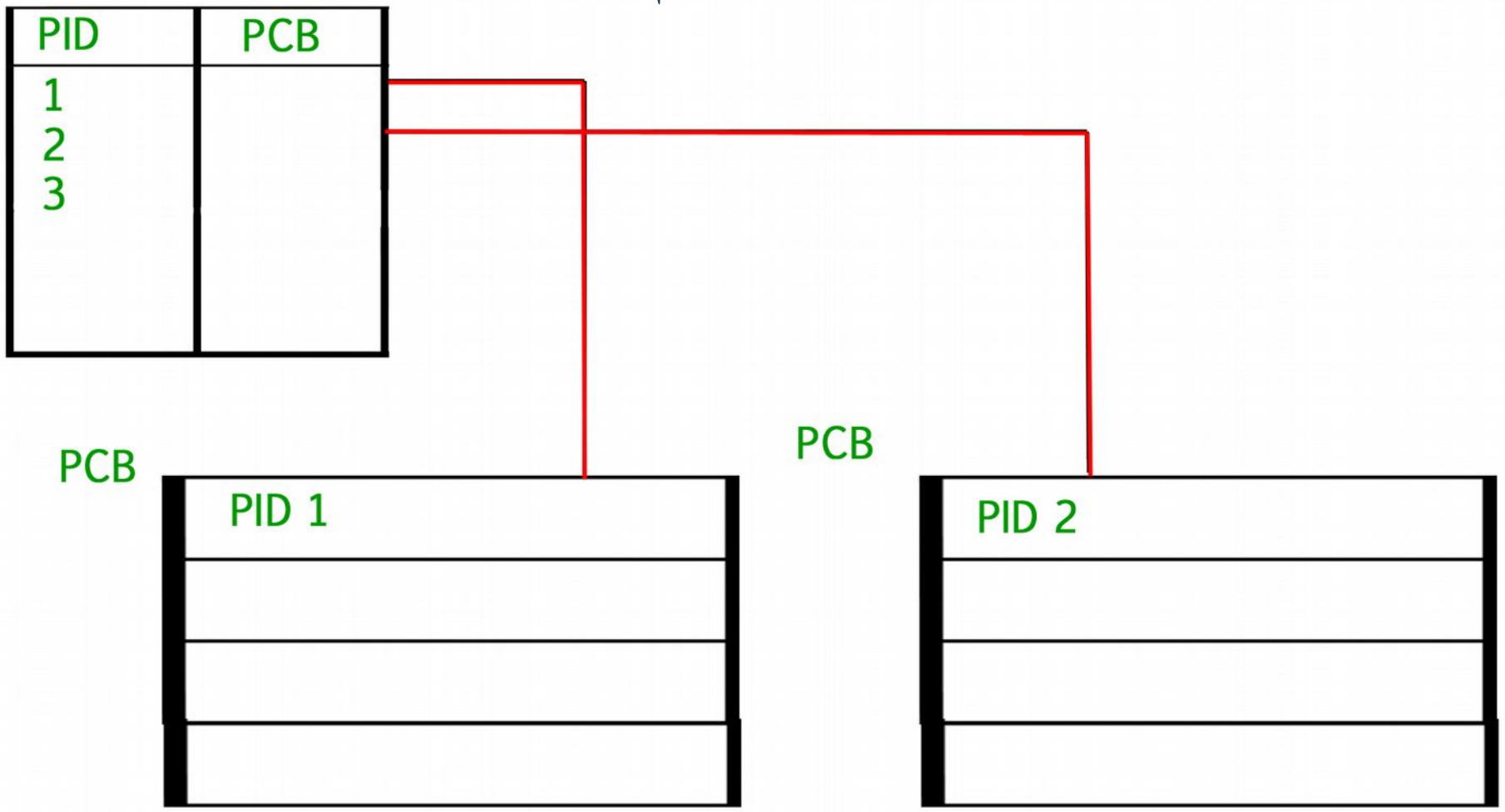
بلوک کنترل فرآیند Process Control Block



از اسلایدهای کتاب سیلبرشاتز

جدول فرآیندها Process Table

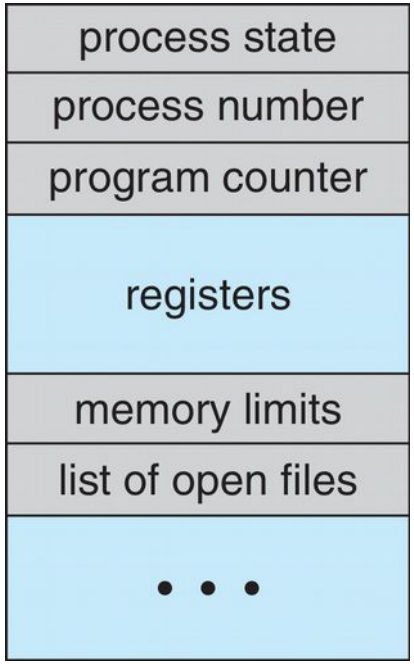
- جدولی از فرآیندهای موجود در سیستم‌عامل



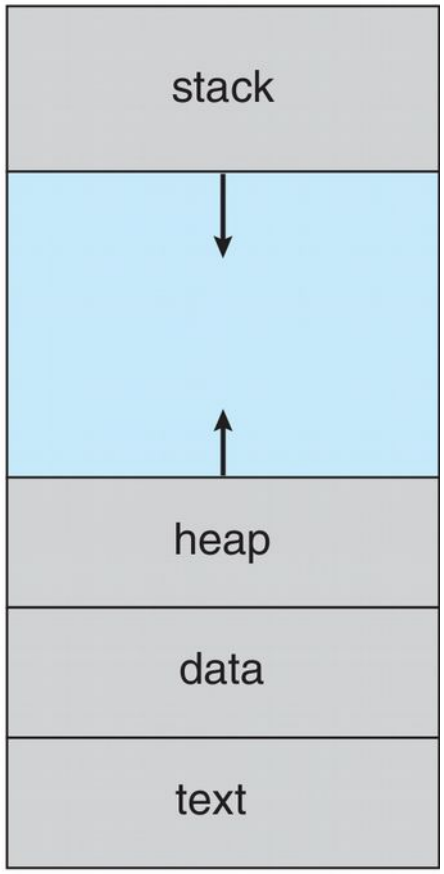
Process table and process control block

بلوک کنترل فرآیند Process Control Block

با داشتن این دو مورد،
می‌توان فرآیند را
متوقف کرد
و دوباره اجرا کرد.

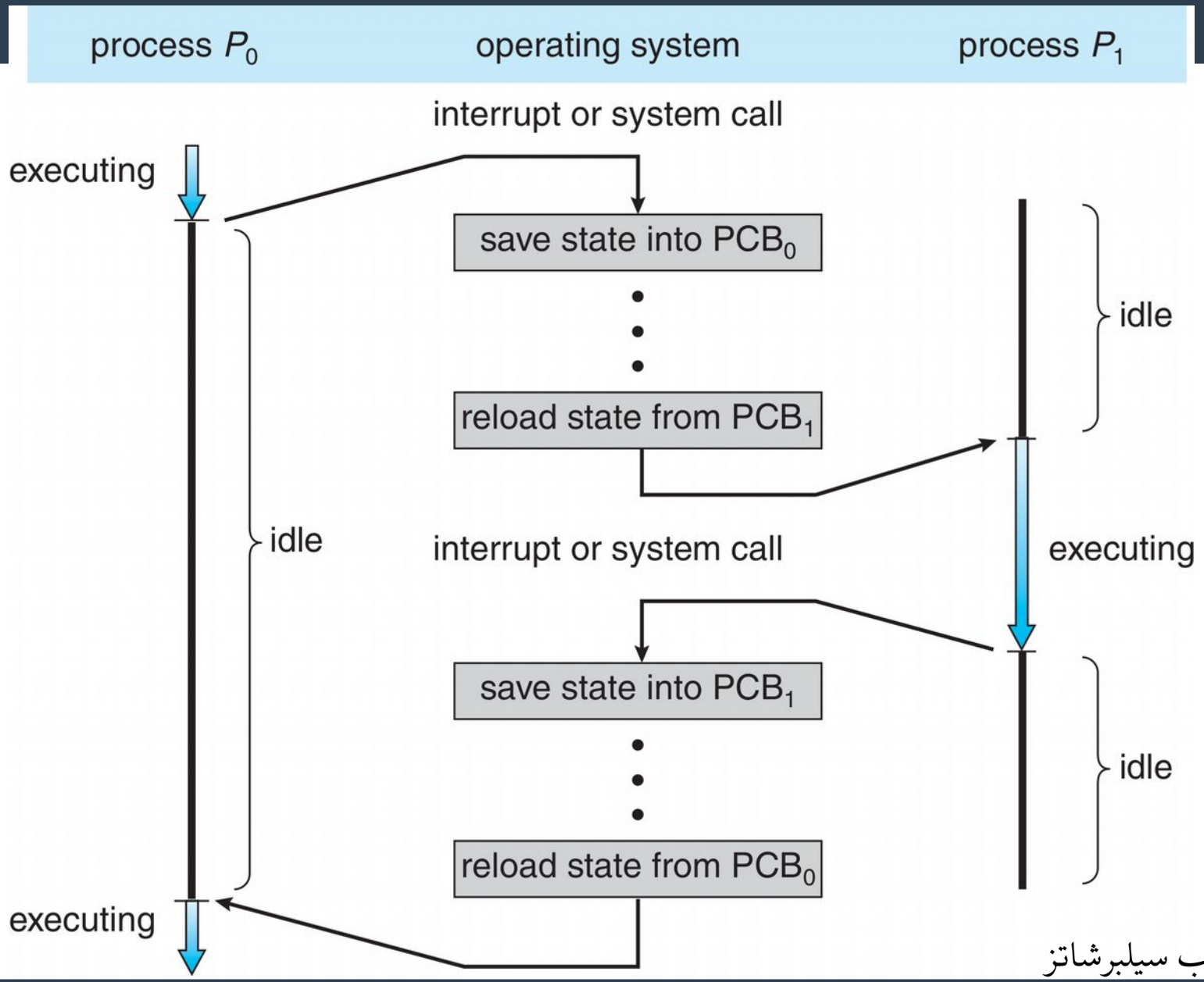


در حافظه سیستم‌عامل



حافظه برنامه

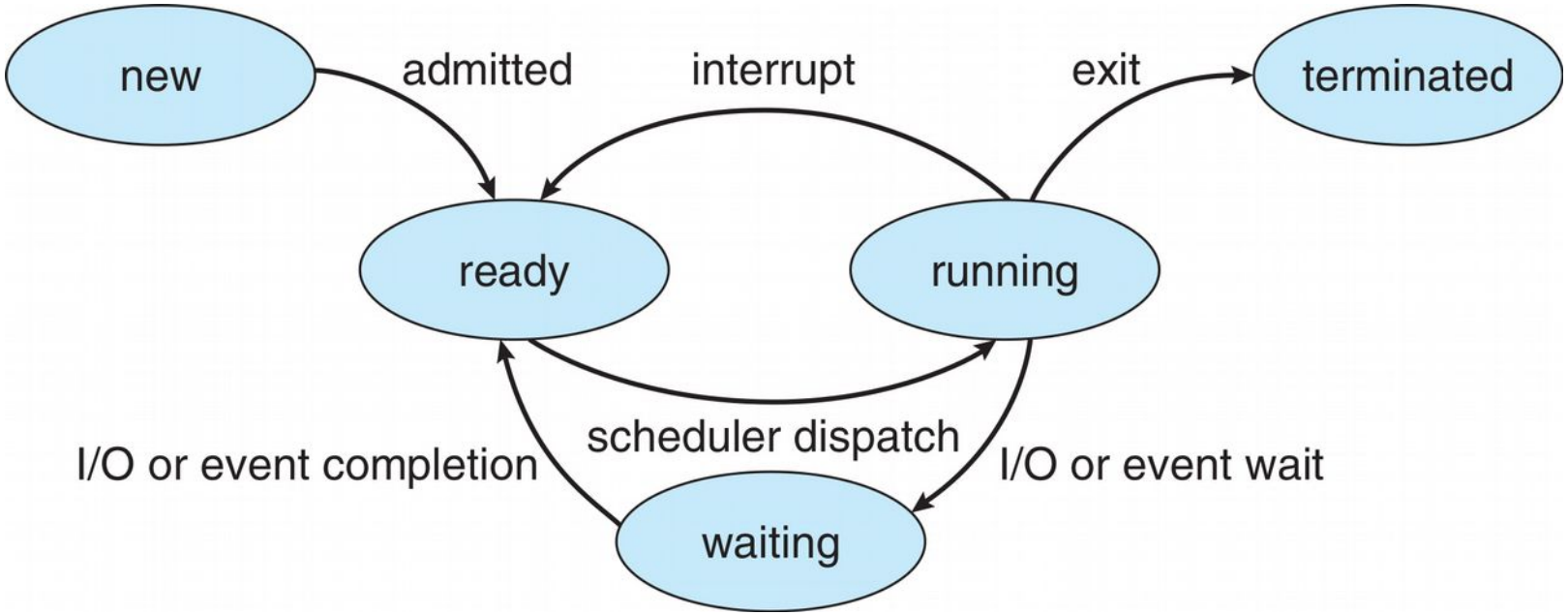
نحوه جابه‌جایی فرآیندها بر روی پردازنده



از اسلایدهای کتاب سیلبرشاتز

زمان‌بندی فرآیند Process Scheduling

- تعیین اینکه یک فرآیند به چه صورتی اجرا بشود.
- چگونه وضعیت فرآیند از **new** به **terminated** می‌رسد.
  - وقتی چند فرآیند در سیستم داریم.

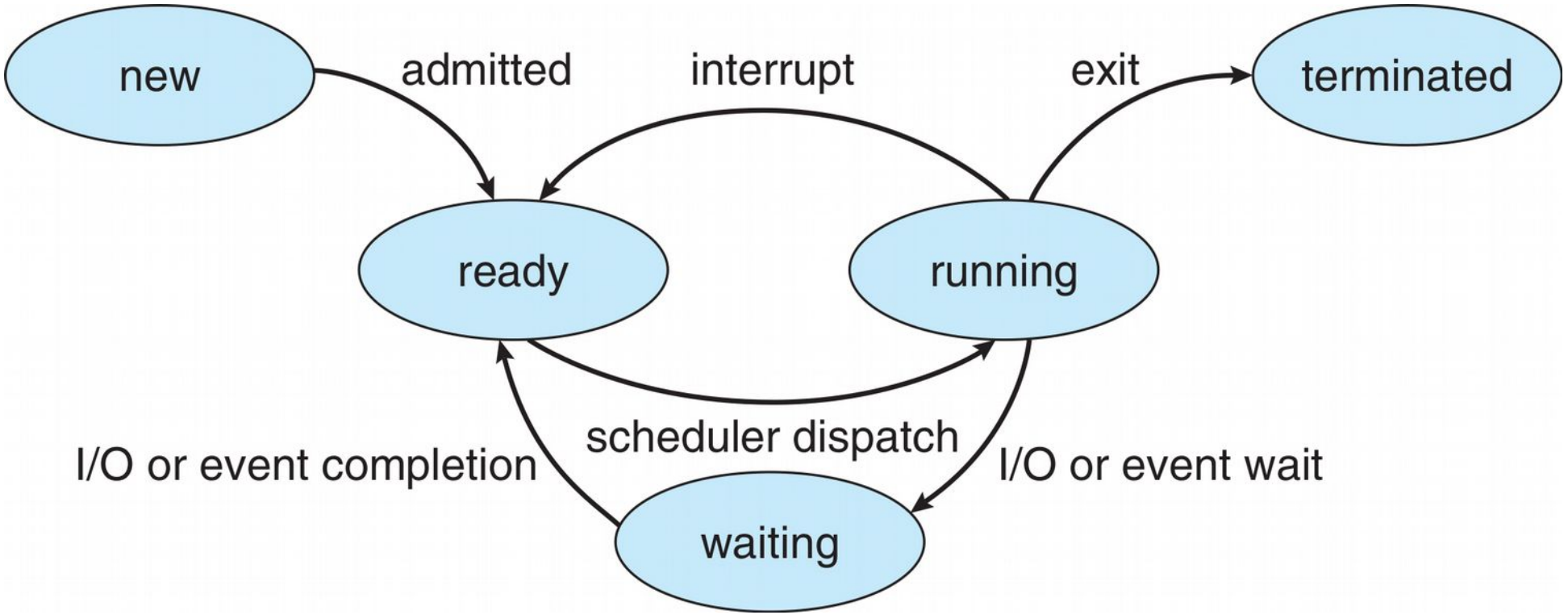


زمان‌بند فرآیند Process Scheduler

- بخشی از (هسته) سیستم‌عامل که عمل زمان‌بندی فرآیند را انجام می‌دهد.
- با استفاده از یک سری صف زمان‌بندی (**Scheduling Queue**) اینکار را انجام می‌دهد.
- نمودار وضعیت فرآیند، برای همه فرآیندها یکسان است.
- هر فرآیند در هر لحظه دارای یک وضعیت است.
- سؤال: در هر لحظه از زمان، چند فرآیند دارای وضعیت **x** است؟

زمان‌بند فرآیند Process Scheduler

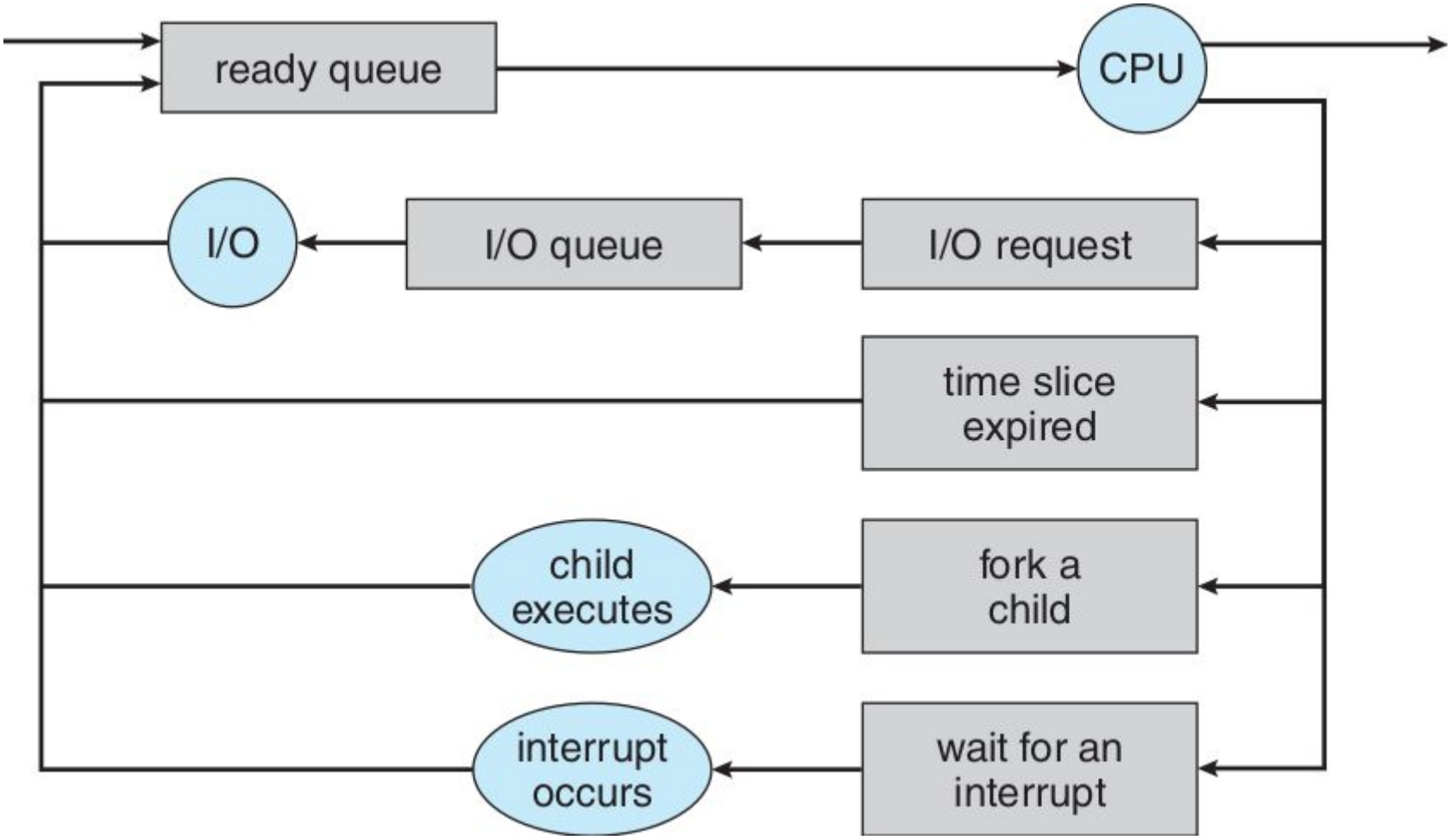
- سؤال: در هر لحظه از زمان، چند فرآیند دارای وضعیت **x** است؟



صف‌های زمان‌بندی Scheduling Queues

- در هر لحظه از زمان، در برخی از وضعیت‌ها، چند فرآیند وجود دارد.
- خدمات‌دهی به این فرآیندها نیاز به یک سری صف دارد.
- صف‌های زمان‌بندی، یک صف‌هایی هستند که فرآیندها در آن قرار می‌گیرند.
  - اشاره‌گر به PCB فرآیند
  - بخشی از سیستم‌عامل

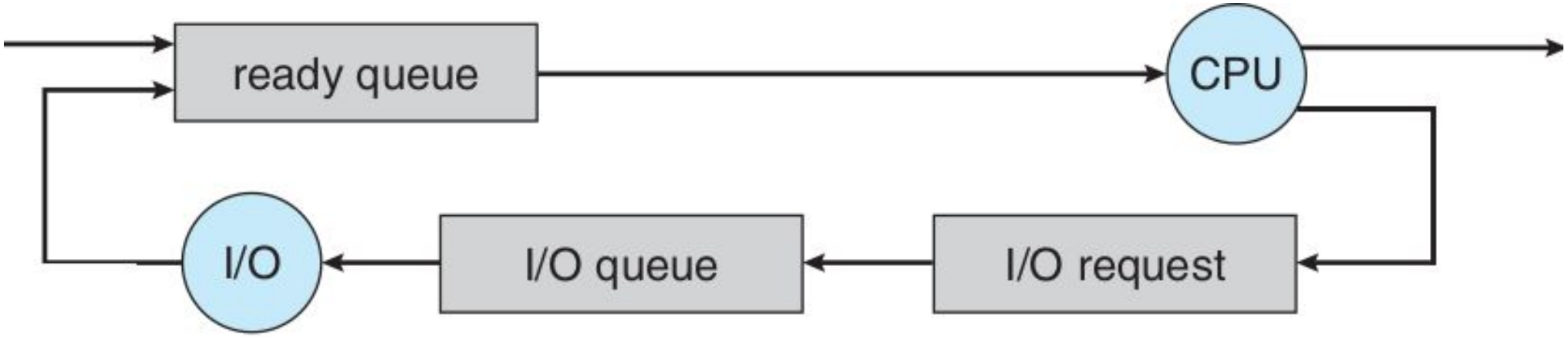
صف‌های زمان‌بندی Scheduling Queues



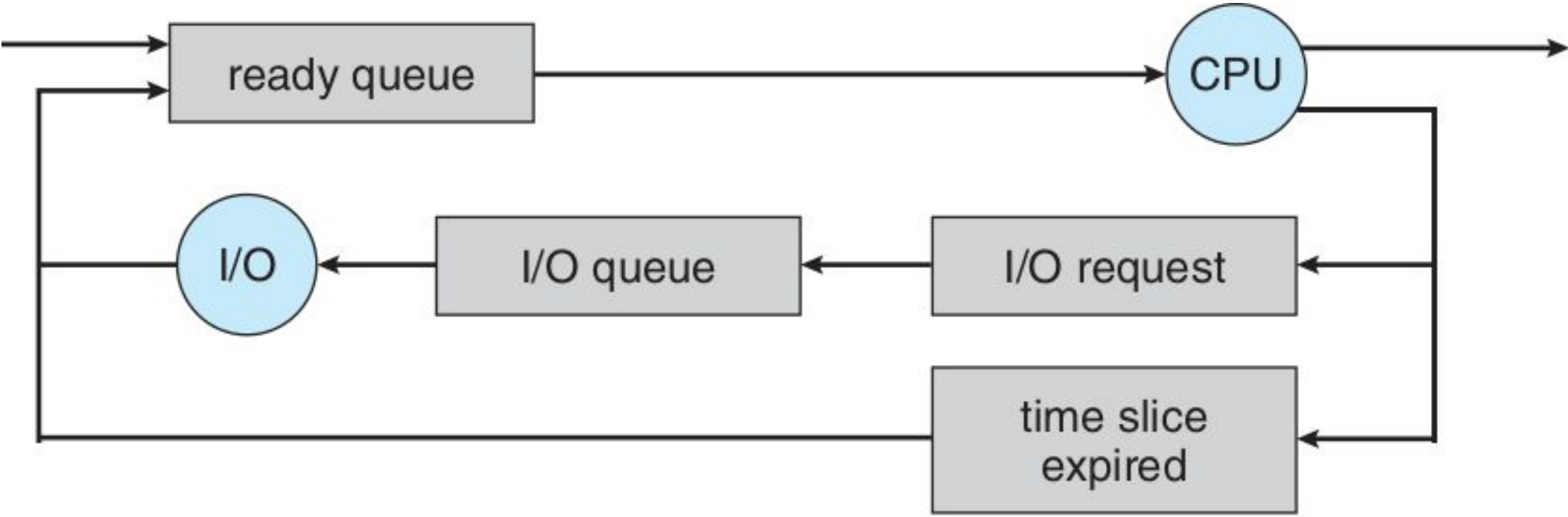
از اسلایدهای کتاب سیلبرشاتز

صف‌های زمان‌بندی Scheduling Queues

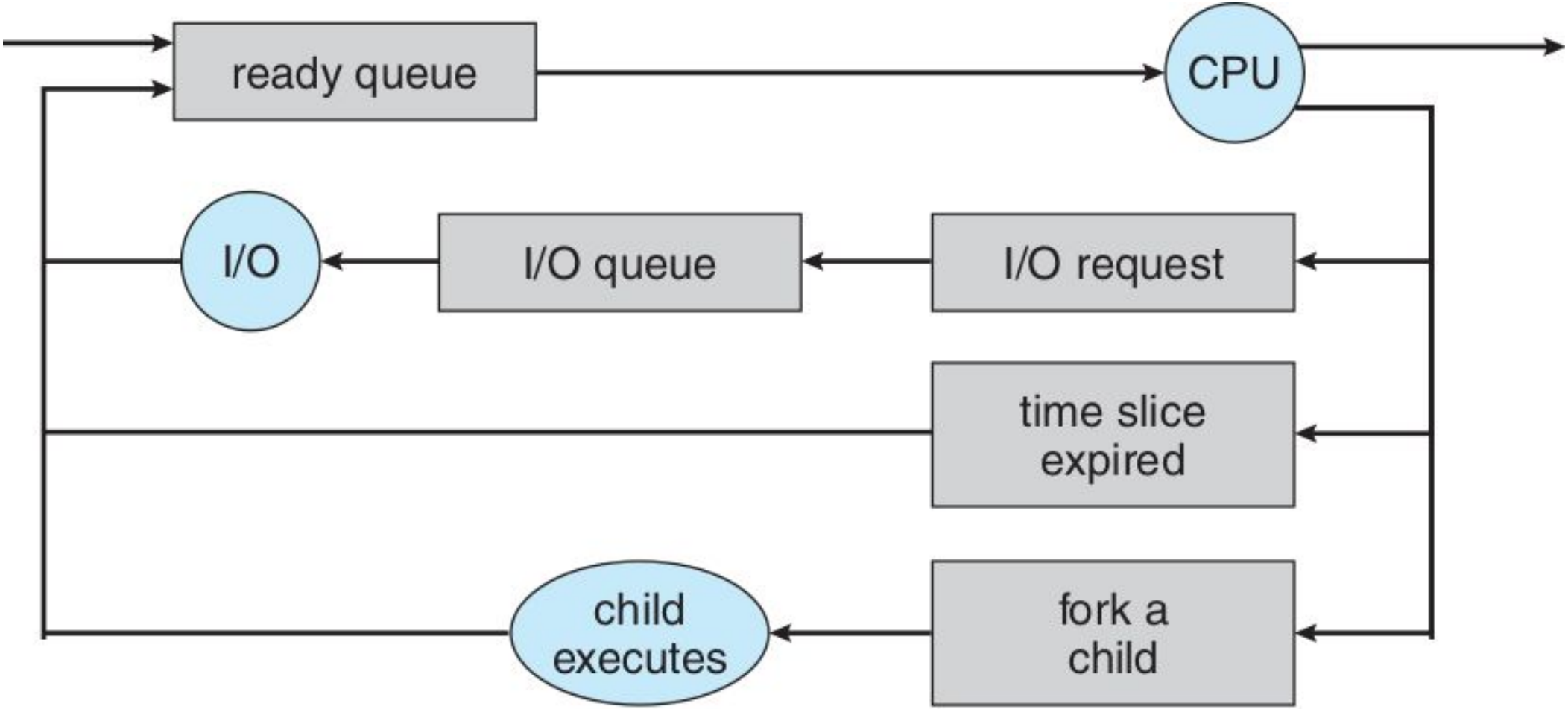
صف‌های زمان‌بندی Scheduling Queues



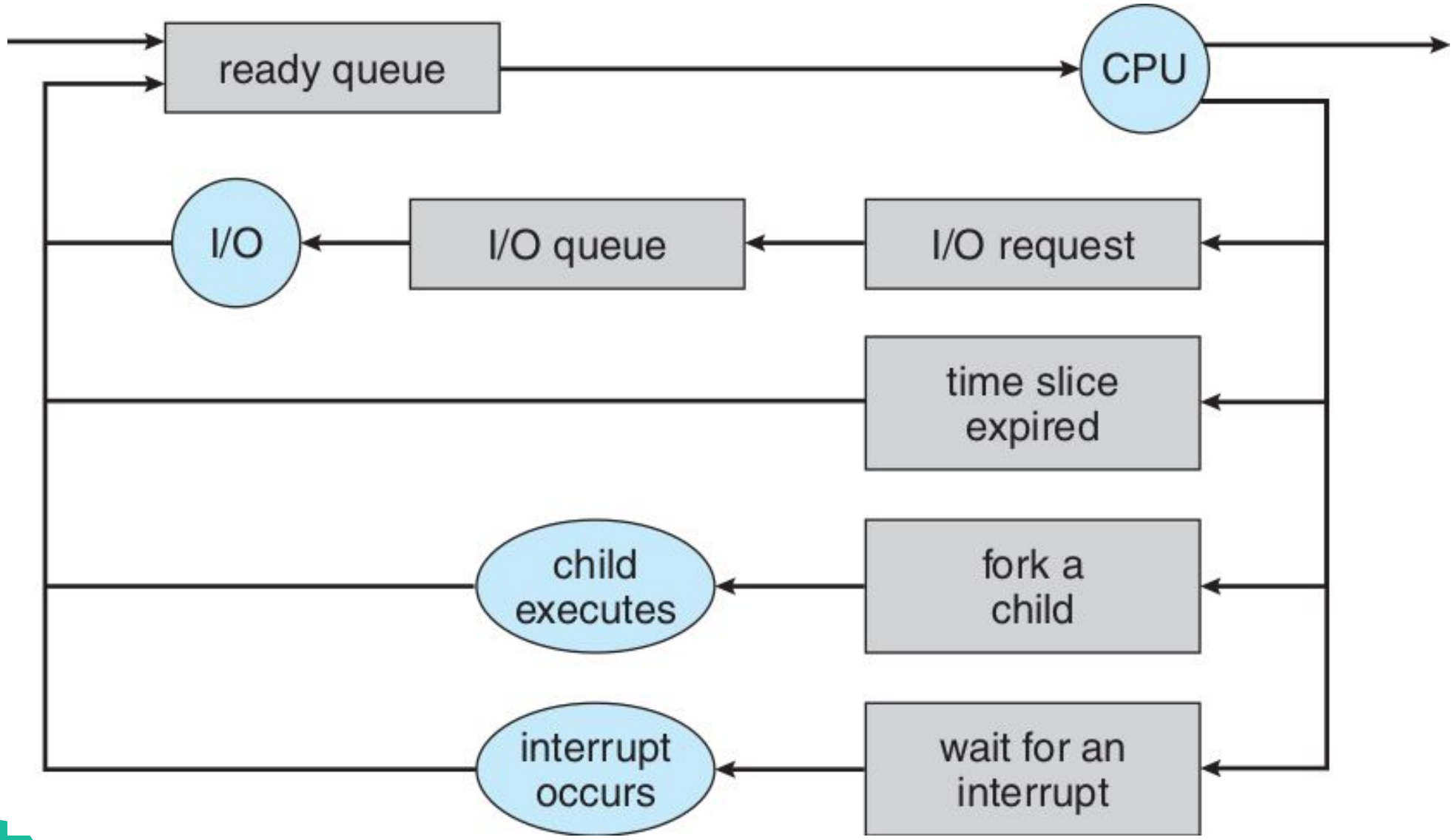
صف‌های زمان‌بندی Scheduling Queues



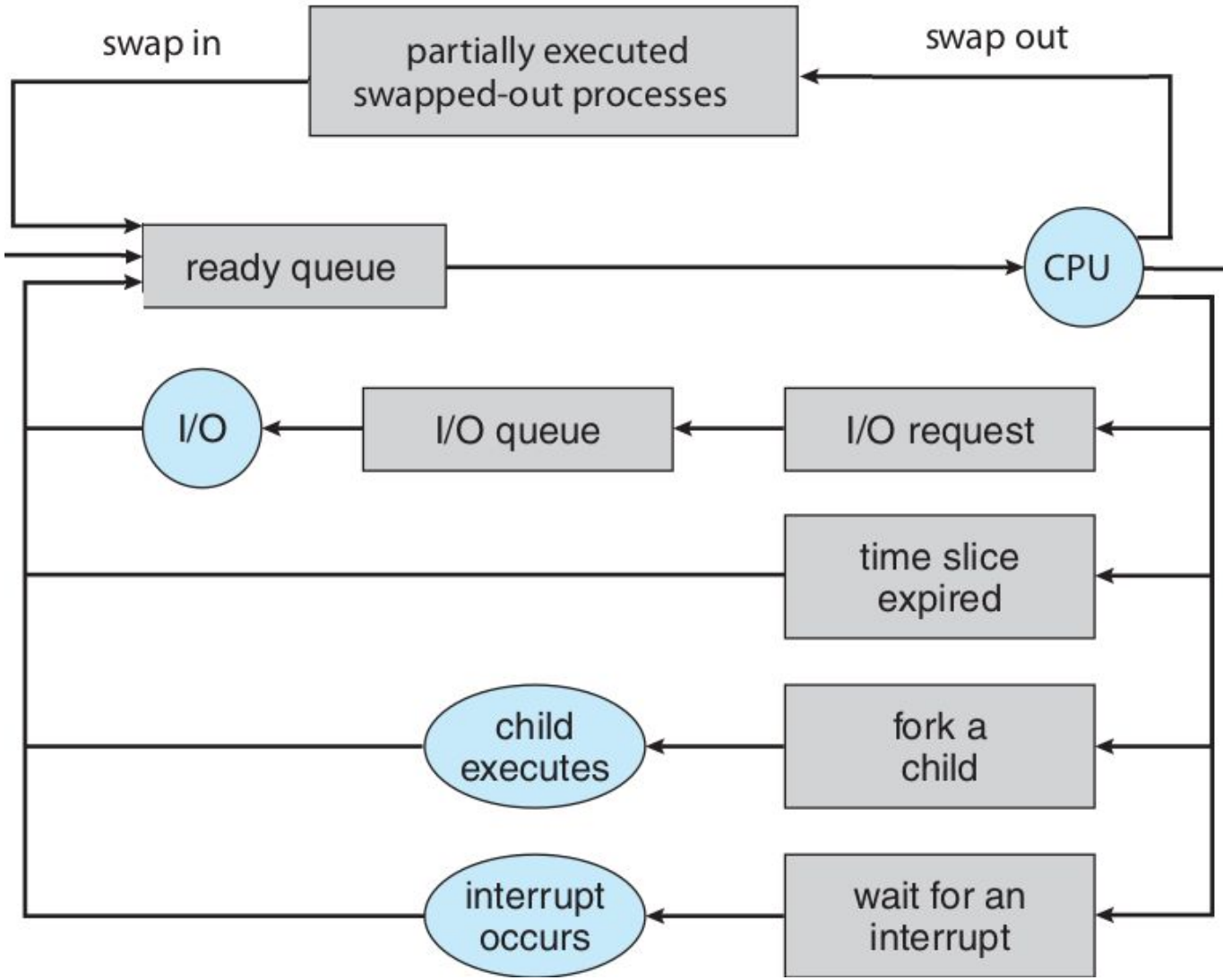
صف‌های زمان‌بندی Scheduling Queues



صف‌های زمان‌بندی Scheduling Queues

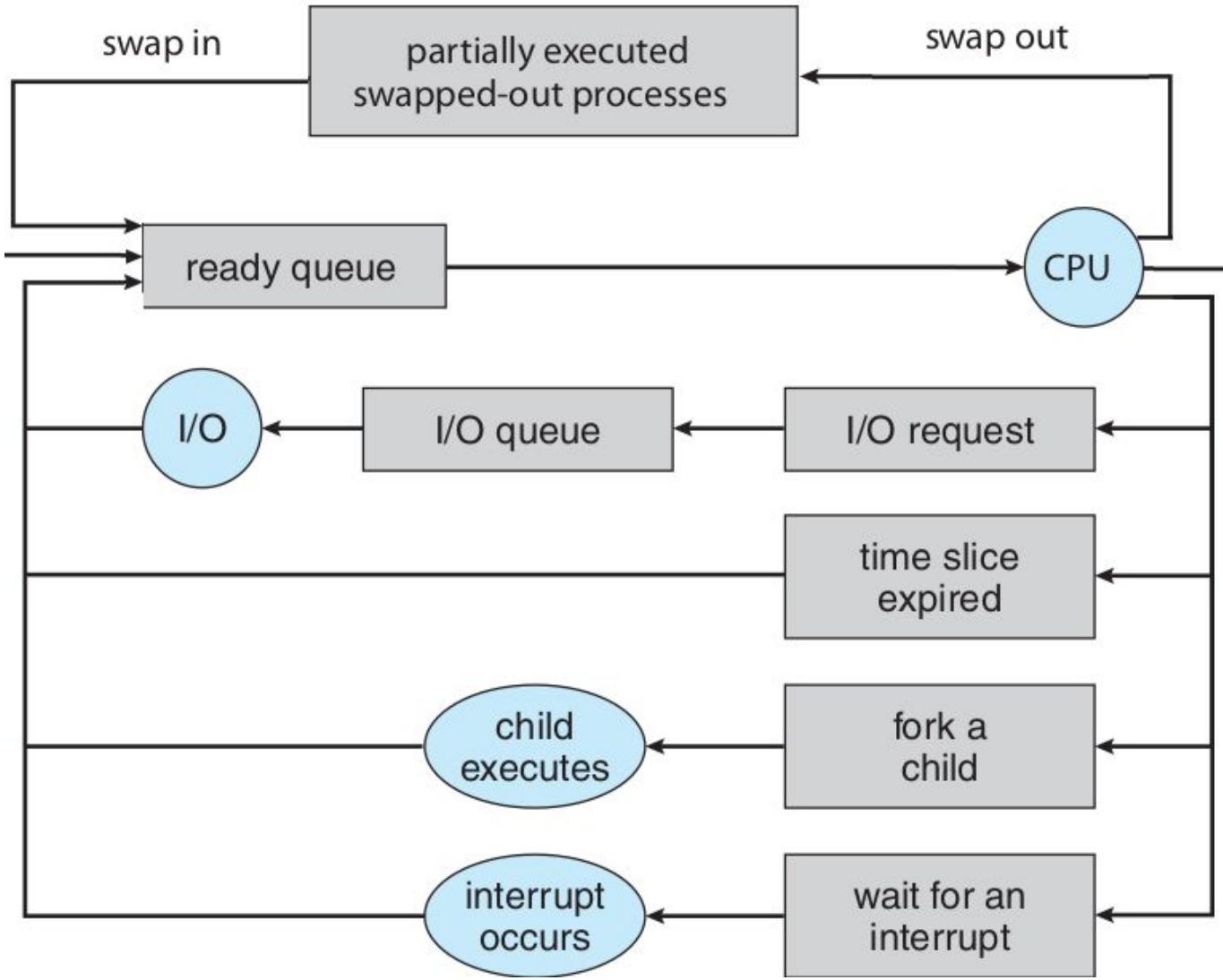


صف‌های زمان‌بندی Scheduling Queues



- مبادله Swap
- خالی کردن حافظه
- کم کردن بار روی CPU

صف‌های زمان‌بندی Scheduling Queues



- مبادله Swap
- خالی کردن حافظه
- کم کردن بار روی CPU

از اسلایدهای کتاب سیلبرشاتز

تغییر متن Context Switch

- پردازنده چطور فرآیندهای مختلف را اجرا می‌کند؟
- فرض کنید پردازنده در حال اجرای فرآیند **A** است.
- پس از آن، باید فرآیند **B** اجرا شود.
- همانطور که قبلاً گفتیم، برخی از اطلاعات فرآیند در پردازنده وجود دارد
  - رجیستر PC (آدرس دستور بعدی که باید اجرا شود)
  - رجیسترهای همه منظوره
  - رجیسترهای کنترلی

تغییر متن Context Switch

- برخی از اطلاعات فرآیند در پردازنده وجود دارد
  - وضعیت پردازنده CPU State
- قبل از اجرای فرآیند **B** بر روی پردازنده، باید اطلاعات فرآیند **A** که در پردازنده وجود دارد، ذخیره شود.
  - در کجا؟ در PCB فرآیند A
  - به اینکار ذخیره وضعیت State Save گفته می‌شود.
- هنگامی که خواستیم اجرای **A** را از سر بگیریم، باید این اطلاعات را بر روی پردازنده قرار دهیم
  - اطلاعات را از کجا بیاوریم؟ از PCB فرآیند A
  - به اینکار بازگشت وضعیت State Restore گفته می‌شود.

تغییر متن Context Switch

- **Context Switch**:
  - عملیاتی که طی آن یک فرآیند از روی پردازنده برداشته می‌شود و فرآیند دیگر بر روی پردازنده قرار می‌گیرد.
- دارای دو عمل است
  - State Save
    - برای فرآیندی که هم‌اکنون بر روی پردازنده اجرا می‌شود
  - State Restore
    - برای فرآیندی که قرار است بر روی پردازنده اجرا شود
- یک سرباز است و در این زمان کار مفیدی انجام نمی‌شود.
- پردازنده‌های جدید، دارای دستورالعمل‌های مخصوصی برای اینکار هستند.