



دانشگاه صنعتی شاهرود  
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

# سیستم‌های عامل

## عملیات روی فرآیندها – ارتباط بین فرآیندها

استاد: علیرضا تجری

# عملیات روی فرآیندها

- ایجاد فرآیند

- رابطه Parent و Child

- درخت فرآیندها

- فراخوانی‌های سیستمی

- بستن (اتمام) فرآیند

- رابطه Parent و Child

- فرآیند Zombie و فرآیند Orphan

- فرآیندها در جاوا

# عملیات روی فرآیندها

## • ایجاد فرآیند

### - رابطه Parent و Child

- یک فرآیند جدید توسط یک فرآیند دیگر ایجاد می‌شود.
  - شما هم می‌توانید با برنامه نویسی، یک فرآیند جدید ایجاد کنید.
- فرآیندها دارای رابطه Parent-Child هستند.
  - فرآیند ایجاد کننده Parent
  - فرآیند ایجاد شده Child
- در لینوکس در ابتدا یک فرآیند ایجاد می‌شود و آن فرآیند، فرآیندهای دیگر را ایجاد می‌کند.
  - در نسخه‌های قدیمی init و در نسخه‌های جدید systemd

# عملیات روی فرآیندها

- ایجاد فرآیند

– به دلیل رابطه Parent و Child، می‌توانیم یک درخت رسم کنیم.

- درخت فرآیند

- دستور pstree در لینوکس

– اسلاید بعد

– ایجاد فرآیند توسط فراخوانی سیستمی انجام می‌شود

- در لینوکس fork

- در ویندوز CreateProcess

– به هر فرآیند یک شناسه اختصاص داده می‌شود.

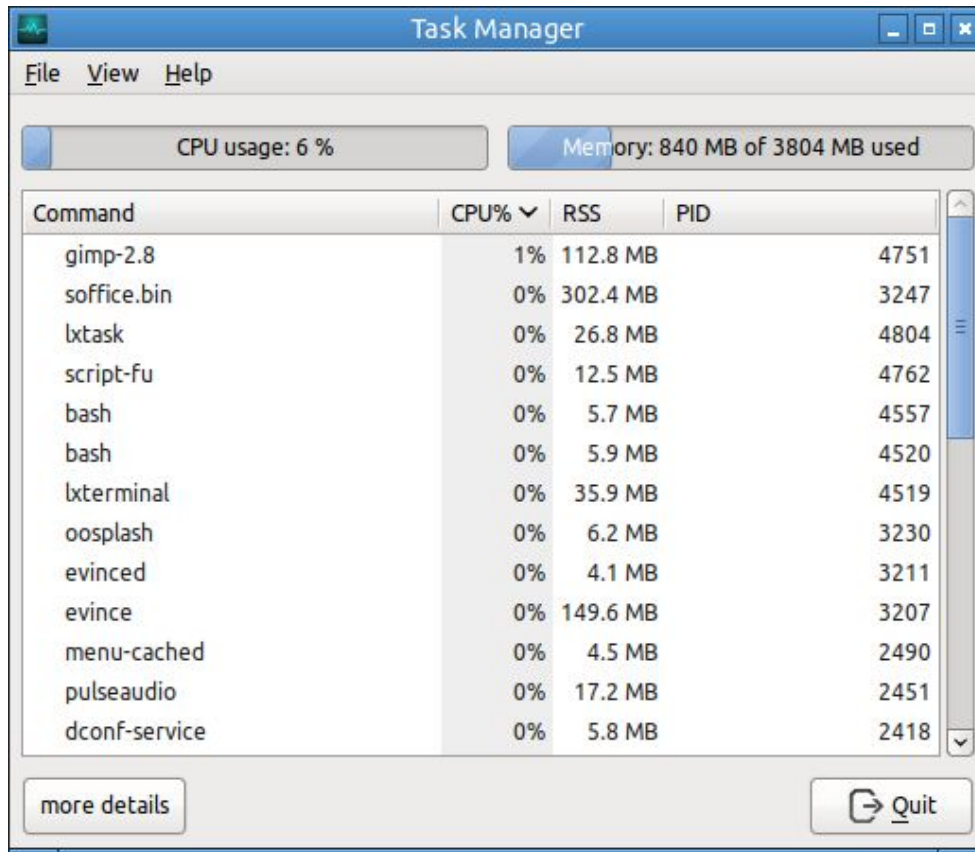
- pid

# عملیات روی فرآیندها / دستور pstree

```
systemd--NetworkManager--dhclient
                        2*[{NetworkManager}]
--accounts-daemon--2*[{accounts-daemon}]
--acpid
--agetty
--atd
--avahi-daemon--avahi-daemon
--cron
--cups-browsed--2*[{cups-browsed}]
--cupsd
--dbus-daemon
--dnsmasq
--dnsmasq--dnsmasq
--irqbalance--{irqbalance}
--2*[{kerneloops}]
--libvirtd--16*[{libvirtd}]
--light-locker--3*[{light-locker}]
--lightdm--Xorg--11*[{Xorg}]
           |
           |--lightdm
           |
           |--lxsession--lxpanel--lxterminal--bash--pstree
           |                  |
           |                  |--bash
           |                  |--2*[{lxterminal}]
           |                  |--3*[{lxpanel}]
           |                  |--openbox
           |                  |--pcmanfm--evince--4*[{evince}]
           |                  |
           |                  |--oosplash--soffice.bin--4*[{soffice.bin}]
           |                  |
           |                  |--2*[{oosplash}]
           |                  |--2*[{pcmanfm}]
           |                  |--ssh-agent
           |                  |--today--2*[{today}]
           |                  |--2*[{lxsession}]
           |                  |--2*[{lightdm}]
           |                  |--2*[{lightdm}]
```

# عملیات روی فرآیندها

- فراخوانی‌های سیستمی دیگری نیز وجود دارد
  - مثلاً به دست آوردن حجم حافظه استفاده شده و ...
- یک پارامتر اصلی: pid



The screenshot shows the Windows Task Manager window. At the top, it displays 'CPU usage: 6 %' and 'Memory: 840 MB of 3804 MB used'. Below this is a table of running processes with columns for Command, CPU%, RSS, and PID. The processes listed include gimp-2.8, soffice.bin, lxtask, script-fu, bash, lxterminal, oosplash, evince, menu-cached, pulseaudio, and dconf-service. At the bottom, there are buttons for 'more details' and 'Quit'.

Command	CPU%	RSS	PID
gimp-2.8	1%	112.8 MB	4751
soffice.bin	0%	302.4 MB	3247
lxtask	0%	26.8 MB	4804
script-fu	0%	12.5 MB	4762
bash	0%	5.7 MB	4557
bash	0%	5.9 MB	4520
lxterminal	0%	35.9 MB	4519
oosplash	0%	6.2 MB	3230
evince	0%	4.1 MB	3211
evince	0%	149.6 MB	3207
menu-cached	0%	4.5 MB	2490
pulseaudio	0%	17.2 MB	2451
dconf-service	0%	5.8 MB	2418



Process Explorer

# عملیات روی فرآیندها

- بستن (اتمام) فرآیند

- با فراخوانی سیستمی `exit`، یک فرآیند خاتمه می‌یابد.
- یک فرآیند دیگر هم می‌تواند این فرآیند را خاتمه بدهد.

- با فراخوانی سیستمی `kill` و ارسال سیگنال

- رابطه Parent و Child

- وقتی child بسته شد (خاتمه یافت - وضعیت Terminated)، چه اتفاقی می‌افتد؟
- وقتی Parent بسته شد، چه اتفاقی برای Child می‌افتد؟

# عملیات روی فرآیندها

- بستن (اتمام) فرآیند

- رابطه Parent و Child

- وقتی child بسته شد، چه اتفاقی می افتد؟

- منابع مربوط به فرآیند آزاد می شود (مثل حافظه اختصاص داده شده، فایل ها و پورت های شبکه باز شده).

- مقدار exit code در PCB موجود است.

- داده مهمی است. (وضعیت اجرای فرآیند: 0 اجرای بدون مشکل)

- ممکن است Parent به مقدار آن نیاز داشته باشد.

- در این حالت اصطلاحاً فرآیند Zombie است.

- هنگامی که فرآیند Parent، فراخوانی سیستمی wait را اجرا می کند، مقدار exit code را دریافت می کند و PCB فرآیند حذف می شود.



# عملیات روی فرآیندها

- بستن (اتمام) فرآیند

- رابطه Parent و Child

- وقتی Parent بسته شد، چه اتفاقی برای Child می افتد ؟

- فرآیندهای Child، اصطلاحاً Orphan می شوند.

- یک فرآیند دیگر، والد آنها می شود.

- معمولاً اولین فرآیند.

# عملیات روی فرآیندها

- نمونه جاوا برای کار با فرآیندها

- ایجاد فرآیند جدید

- `ProcessBuilder.start()`
    - `Runtime.getRuntime().exec()`

- بستن فرآیند ایجاد شده

- `process.destroy();`

- منتظر ماندن برای اتمام اجرای فرآیند ایجاد شده

- `process.waitFor();`

# عملیات روی فرآیندها

- نمونه جاوا برای کار با فرآیندها

```
public class ProcessDemo{  
    public static void main(String[] args){  
        try{  
            Process p = Runtime.getRuntime().exec("notepad.exe");  
            p.waitFor();  
        } catch (Exception ex){  
            ex.printStackTrace();  
        }  
    }  
}
```

# ارتباطات بین فرآیندها

- روابط فرآیندها با هم
- روش‌های اشتراک گذاری داده بین فرآیندها
  - حافظه مشترک
  - ارسال پیام
- ارتباطات در سیستم‌های **Client/Server**
  - سوکت‌های شبکه
  - RPC و RMI
  - Pipe

# ارتباطات بین فرآیندها

- روابط فرآیندها با هم

- فرآیندهای مستقل: اجرای دو فرآیند ارتباطی با هم ندارد.

- نوشتن ایمیل و گوش دادن به موسیقی

- فرآیندهای همکار: دو فرآیند داده‌هایی را بین همدیگر به اشتراک می‌گذارند.

- مثلاً مرورگر که فایل‌هایی را از سرور وب دریافت می‌کند و نشان می‌دهد.

- مکانیزم‌هایی برای ارتباط بین فرآیندها وجود دارد

- IPC: Inter-Process Communication

- ارتباط بین فرآیندها، از طریق به اشتراک گذاری داده است.

# ارتباطات بین فرآیندها

## • روش‌های اشتراک گذاری داده بین فرآیندها

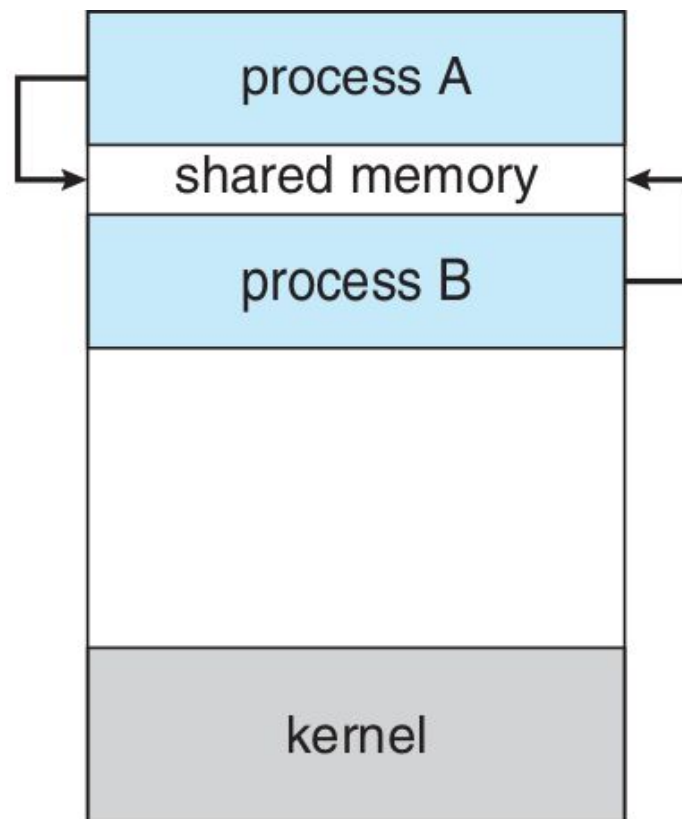
### - حافظه مشترک Shared Memory

- بخشی از حافظه اصلی، بین فرآیندهای همکار به اشتراک گذاشته می‌شود. اینکار نیاز به فراخوانی سیستمی دارد.
- سؤال: فرض کنید دو فرآیند در حال اجرا هستند. آیا نمی‌توانند بخشی از حافظه خود را به اشتراک بگذارند؟
- خیر. هر فرآیند فقط به حافظه خود دسترسی دارد. هر آدرسی که درخواست بدهد، به حافظه خودش برمی‌گردد.
- فقط برای راه‌اندازی، نیاز به فراخوانی سیستمی دارد. پس از آن، هر دو فرآیند، به حافظه مشترک دسترسی دارند (بدون نیاز به فراخوانی سیستمی).

# ارتباطات بین فرآیندها

- روش‌های اشتراک گذاری داده بین فرآیندها

– حافظه مشترک Shared Memory



# ارتباطات بین فرآیندها

## • روش‌های اشتراک گذاری داده بین فرآیندها

### – ارسال پیام Message Passing

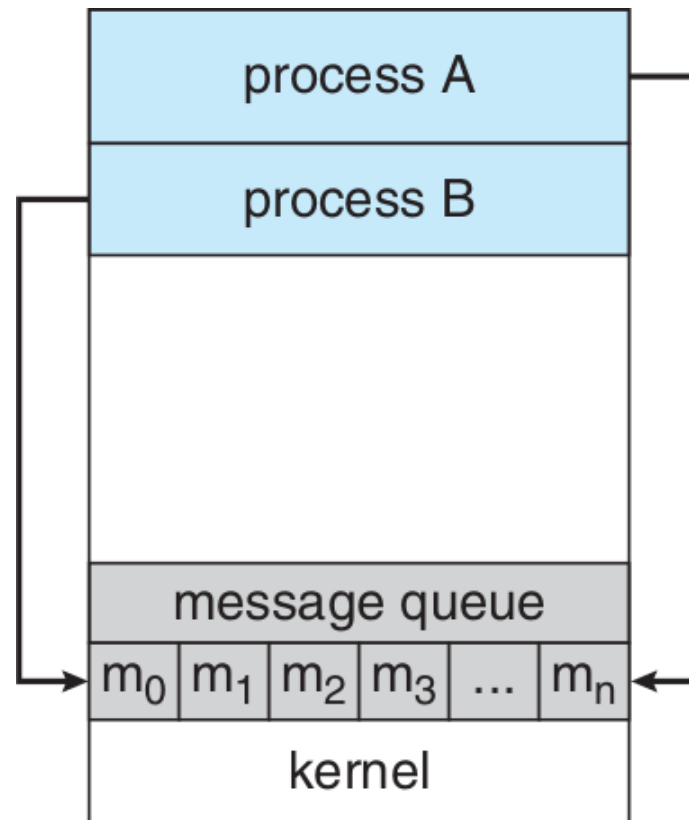
- اشتراک گذاری داده از طریق ارسال پیام انجام می‌شود.
  - هر ارسال و دریافت پیام نیاز به فراخوانی سیستمی دارد.
  - فرستنده، پیام‌ها را در یک صف به هسته سیستم عامل ارسال می‌کند.
  - خواننده، پیام‌ها را از هسته سیستم عامل می‌خواند.
  - در گذشته به دلیل سربار بالا، کمتر استفاده می‌شد.
- اما امروزه با پردازنده‌های چند هسته‌ای دارای Cache، نسبت به قبل، بیشتر استفاده می‌شود.



# ارتباطات بین فرآیندها

- روش‌های اشتراک گذاری داده بین فرآیندها

– ارسال پیام Message Passing



# ارتباطات بین فرآیندها

- ارتباطات در سیستم‌های **Client/Server**

- سوکت‌های شبکه

- سوکت سرور و کلاینت / شماره پورت و آدرس IP

- RMI و RPC

- Remote Procedure Call

- فراخوانی تابع از راه دور

- ارسال پارامترهای تابع و نام تابع از کامپیوتر مبدا به کامپیوتر مقصد و دریافت خروجی از کامپیوتر مقصد.

- Pipe

- پایپ معمولی

- پایپ دارای نام

# ارتباطات بین فرآیندها

- پایپ معمولی

- هر فرآیند پس از ایجاد شدن دارای سه جریان داده است.

- ورودی استاندارد cin

- خروجی استاندارد cout

- خروجی خطا cerr

- می‌توان خروجی استاندارد یک فرآیند را به ورودی استاندارد فرآیند دیگر هدایت کرد.

- مثل عملگر | در شل

ls | wc -l

- فراخوانی سیستمی pipe

سیستم‌های عامل - دانشکده مهندسی کامپیوتر - دانشگاه صنعتی شاهرود

# ارتباطات بین فرآیندها

- پایپ دارای نام

- یک فایل به عنوان حافظه موقت (داده‌ها در حافظه اصلی قرار می‌گیرند و نه دیسک سخت)
- یک فرآیند در آن می‌نویسد.
- فرآیند دیگر از آن می‌خواند.
- به محض خوانده شدن، داده‌ها پاک می‌شوند.

- دستور **mkfifo**