

4

Procese

27 octombrie 2008

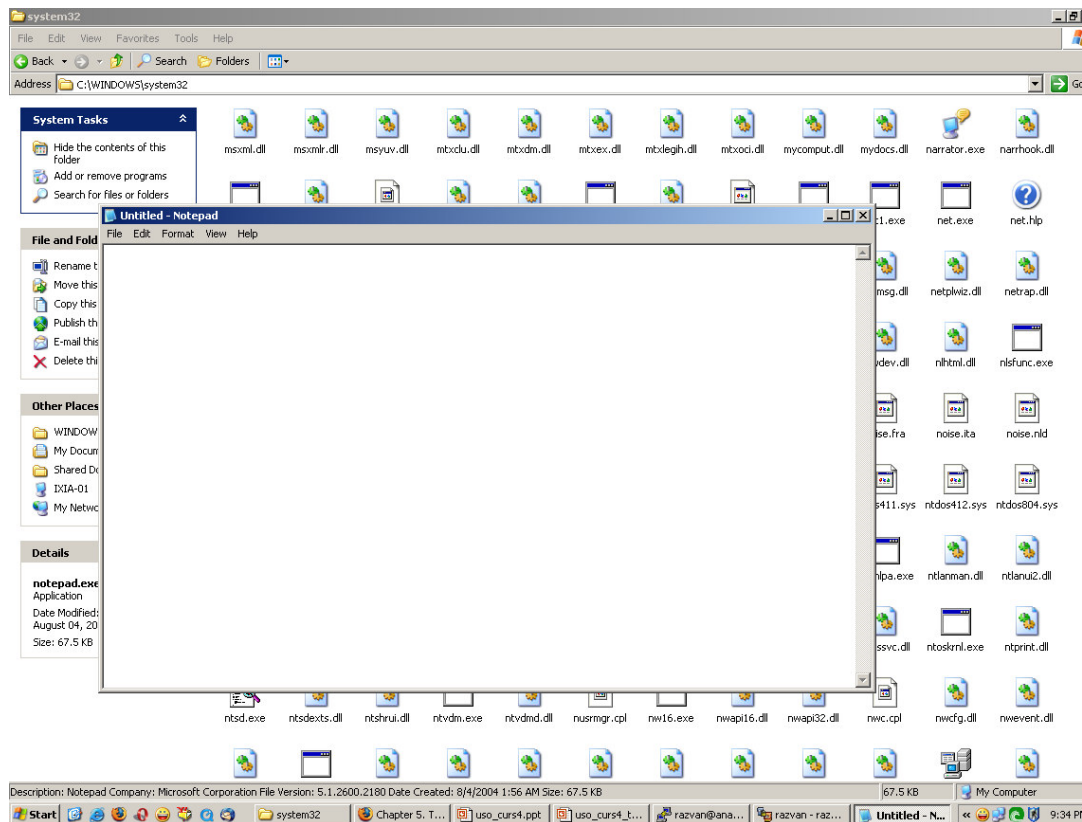
I'm a bit like an operating system really... "multitasking" between these two OHPs, "interrupting" you, ... "sending you to sleep"...

John Bates

- SO single-user până în anii '60
- SO single-user
 - CPU execută o singură instanță de program (până la încheiere)
- Mai mulți utilizatori în același sistem -> mai multe instanțe de programe pentru a fi rulate CPU
- Prima versiune de UNIX avea 2 utilizatori

- Abstractizarea informației despre execuția unui program
 - ce trebuie executat? (program)
 - pentru cât timp? (durată, cuantă)
 - ce condiții vor trebui îndeplinite pentru execuție? (memorie, mecanisme de comunicare, drepturi)
 - pe care procesor? (planificare)

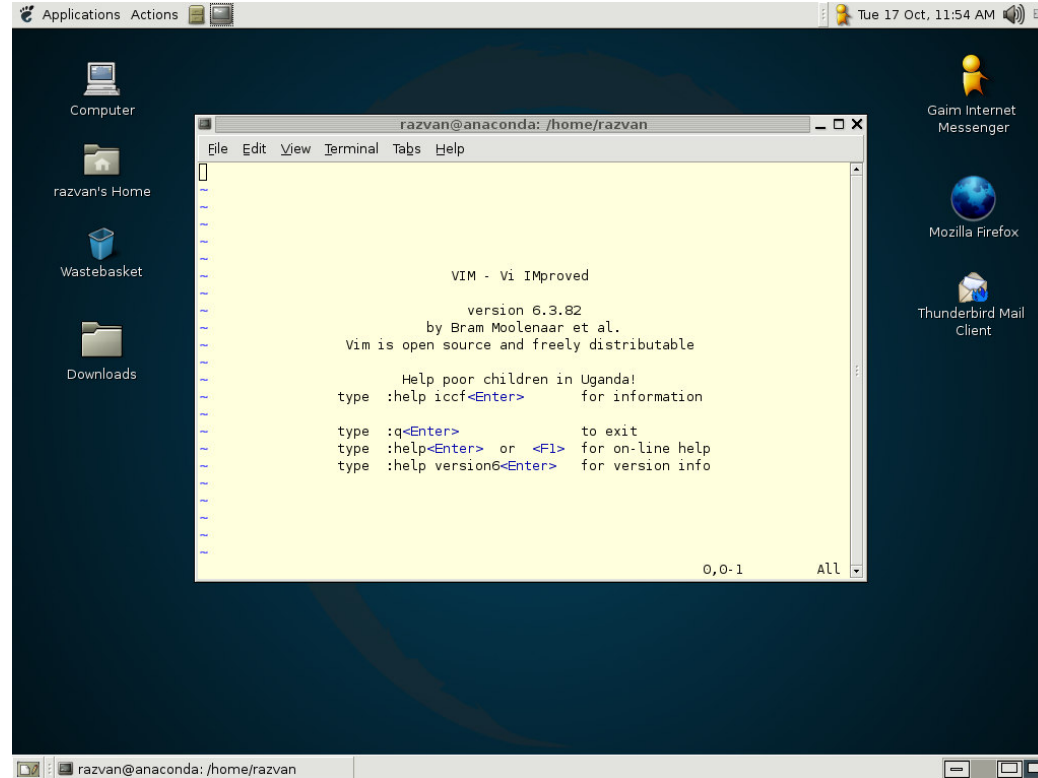
- Este posibilă execuția **simultană** a mai multor instanțe ale unor programe (a mai multor contexte de execuție) pe un sistem de calcul?
- Sisteme multiprocesor
 - folosirea sistemelor de operare multiuser
 - existența simultană a mai multor instanțe de execuție
- Câte programe se pot executa **simultan** pe un sistem multiprocesor?
- De ce, aparent, rulează mai multe instanțe de programe pe un sistem uniprocesor?



```
C:\WINDOWS\system32>dir notepad.exe
Volume in drive C has no label.
Volume Serial Number is D050-E9E7
```

Directory of C:\WINDOWS\system32

```
08/04/2004  04:07 AM                69,120 notepad.exe
             1 File(s)                69,120 bytes
             0 Dir(s)                 453,902,336 bytes free
```



```

razvan@anaconda:~$ whereis vim
vim: /usr/bin/vim.basic /usr/bin/vim [...]
razvan@anaconda:~$ file /usr/bin/vim.basic
/usr/bin/vim.basic: ELF 32-bit LSB executable, Intel 80386,
version 1 (SYSV), for GNU/Linux 2.4.1, dynamically linked (uses
shared libs), for GNU/Linux 2.4.1, stripped
razvan@anaconda:~$ ls -l /usr/bin/vim.basic
-rwxr-xr-x 1 root root 1384232 Aug 29 14:02 /usr/bin/vim.basic

```

- Abstractizare fundamentală în SO (la fel ca și fisierul)
- Un program aflat în execuție (instanță a unui program)
- program vs. proces
 - program
 - entitate pasivă
 - executabil din sistemul de fișiere (aflat pe disc)
 - proces
 - entitate activă
 - instrucțiunile și datele programului se încarcă în memorie și sunt rulate de CPU

- **Unitățile de lucru în SO**
- Dețin informații despre
 - cum se rulează un program în SO
 - când trebuie rulate părți de cod din cadrul unui program
 - cu ce resurse trebuie rulat un program
- Un proces este programul căruia i s-a asociat un context de execuție
- Denumit și task sau job

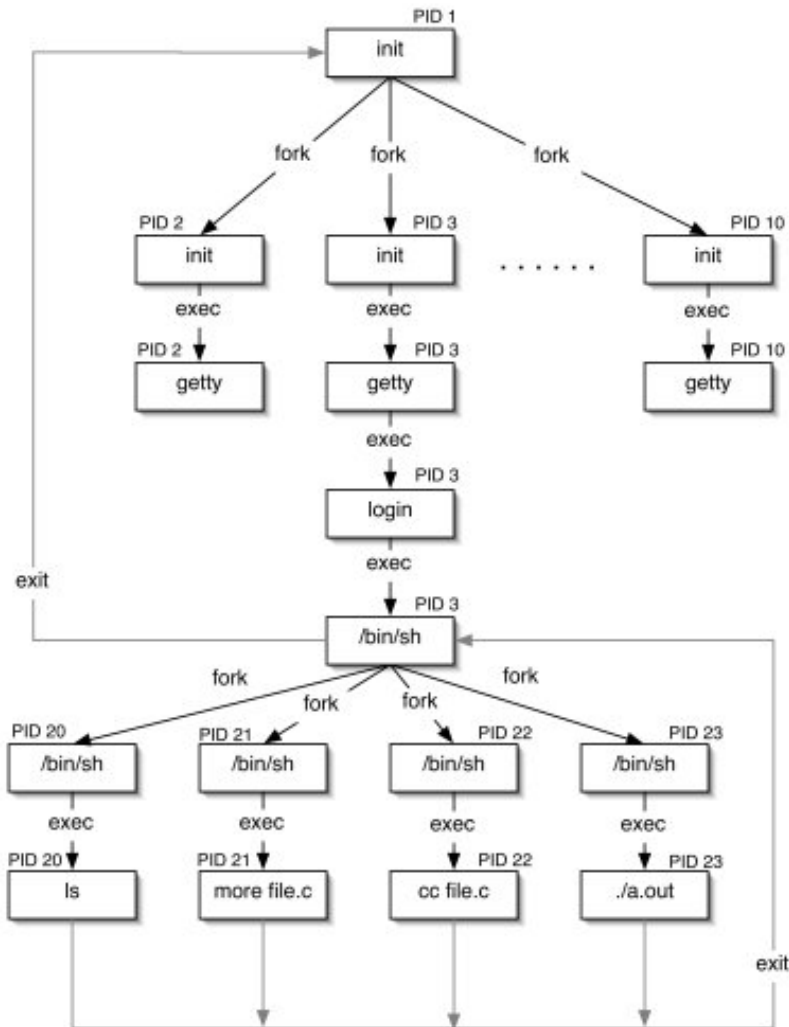
- Situație
 - un singur procesor (sau număr limitat de procesoare)
 - mai multe procese (task-uri)
- Observație: dispozitivele periferice sunt mult mai lente – accesarea unui astfel de dispozitiv durează mult
- Cum se poate realiza planificarea (programarea) proceselor pe procesor?
 - http://en.wikipedia.org/wiki/Computer_multitasking

- Funcționare
 - se alege pentru rulare un proces P1
 - când procesul P1 accesează un dispozitiv periferic, se pune într-o stare specială (sleep, blocat)
 - se planifică alt proces P2 non-blocat
 - totul se reia (salt la primul pas) până când nu mai avem procese
- Avantaje:
 - mărirea eficienței
- Dezavantaje
 - lipsa interactivității

- Se asociază fiecarui proces o cantă de timp (slice)
 - Time slice – timp în care procesul va rula pe procesor
 - Se partejează timpul procesorului – time-sharing
- După expirarea cuantei de timp, procesul va permite **voluntar** altui proces să ruleze pe procesor
- Avantaje:
 - interactivitate
- Dezavantaje:
 - un proces “agățat” (hung, crashed) poate duce la “înghețarea” întregului sistem

- Evoluția multitasking-ului cooperativ
- Folosit preponderent în SO moderne
- Multitasking cooperativ vs. multitasking preemptiv
 - cel preemptiv forțează procesul să cedeze procesorul la expirarea cuantei de timp
 - cooperativ = cedează **voluntar** controlul procesorului;
preemptiv = cedează **obligatoriu** controlul procesorului
- Maximum de interactivitate
- Denumirea actuală de multitasking se referă la un sistem time-sharing, multitasking preemptiv

- Context switch
- Schimbarea execuției unui proces cu un altul pe procesor
- Când apare un context switch?
 - când un proces moare
 - când un proces inițiază comunicare cu un dispozitiv periferic
 - când unui proces îi expiră cuanta de timp
- Planificatorul de procese alege următorul proces
- Planificatorul de procese
 - subsistem al SO
 - algoritmi care conduc la alegerea următorului proces după un context switch



- Identificatorul procesului - **PID** (process ID)
- Orice proces are un proces parinte (și numai unul)
- Un proces este creat prin intermediul unui alt proces
- Ierarhie de procese în Unix
- Procesul **init** – rădăcina ierarhiei

- UNIX

- **init**: părintele tuturor proceselor; responsabil cu pornirea proceselor **daemon** și cu stabilirea contextului de lucru pentru utilizatori
- **swapper**: mutarea proceselor din memorie pe disc (swapping)

- Windows

- **smss.exe** – session manager; primul proces creat; stabilirea contextului pentru utilizatori
- **winlogon.exe** – logon process; se ocupă cu login/logout
- **lsass.exe** – local security authentication server
- **service control manager** – pornirea, oprirea și gestionarea serviciilor

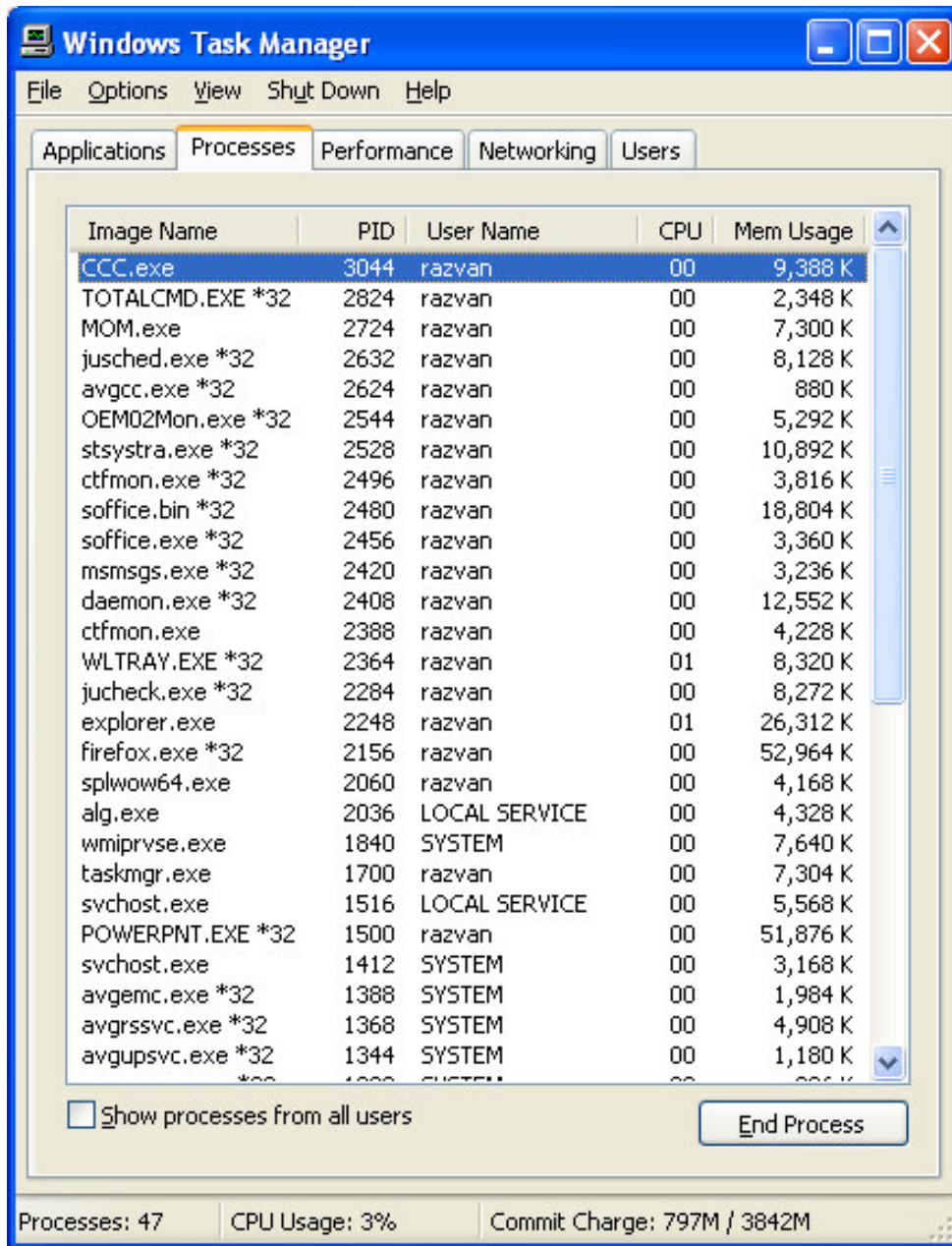


Image Name	PID	User Name	CPU	Mem Usage
CCC.exe	3044	razvan	00	9,388 K
TOTALCMD.EXE *32	2824	razvan	00	2,348 K
MOM.exe	2724	razvan	00	7,300 K
jusched.exe *32	2632	razvan	00	8,128 K
avgcc.exe *32	2624	razvan	00	880 K
OEMO2Mon.exe *32	2544	razvan	00	5,292 K
stsysstra.exe *32	2528	razvan	00	10,892 K
ctfmon.exe *32	2496	razvan	00	3,816 K
soffice.bin *32	2480	razvan	00	18,804 K
soffice.exe *32	2456	razvan	00	3,360 K
mmsgs.exe *32	2420	razvan	00	3,236 K
daemon.exe *32	2408	razvan	00	12,552 K
ctfmon.exe	2388	razvan	00	4,228 K
WLTRAY.EXE *32	2364	razvan	01	8,320 K
jucheck.exe *32	2284	razvan	00	8,272 K
explorer.exe	2248	razvan	01	26,312 K
firefox.exe *32	2156	razvan	00	52,964 K
splwow64.exe	2060	razvan	00	4,168 K
alg.exe	2036	LOCAL SERVICE	00	4,328 K
wmiprvse.exe	1840	SYSTEM	00	7,640 K
taskmgr.exe	1700	razvan	00	7,304 K
svchost.exe	1516	LOCAL SERVICE	00	5,568 K
POWERPNT.EXE *32	1500	razvan	00	51,876 K
svchost.exe	1412	SYSTEM	00	3,168 K
avgemc.exe *32	1388	SYSTEM	00	1,984 K
avgrssvc.exe *32	1368	SYSTEM	00	4,908 K
avgupsvc.exe *32	1344	SYSTEM	00	1,180 K

Processes: 47 CPU Usage: 3% Commit Charge: 797M / 3842M

- Task Manager
- ALT-CTRL-DEL sau CTRL-SHIFT-ESC
- Lista cu procesele active din sistem
- Numele executabilului, al utilizatorului, utilizarea procesorului și memoria ocupată (configurabil)
- Terminarea proceselor nedorite

Vizualizare procese în PowerShell

```
PS C:\Documents and Settings\razvan> ps
Handles  NPM(K)  PM(K)  WS(K) VM(M)  CPU(s)  Id ProcessName
-----  -
          93      8    1404    4328    29     0.02   2036 alg
          100     6    3300    5996    56     0.41    628 ati2evxx
[...]
```

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
93	8	1404	4328	29	0.02	2036	alg
100	6	3300	5996	56	0.41	628	ati2evxx
512	0	44	256	2	13.31	4	System
148	11	7712	2348	80	0.73	2824	TOTALCMD
650	172	10056	5536	99	0.95	372	winlogon
131	9	3072	8320	72	0.45	2364	WLTRAY
46	4	1152	2872	40	0.00	904	WLTRYSVC
168	6	2620	7640	39	0.20	1840	wmioprse

```
PS C:\Documents and Settings\razvan> ps firefox
Handles  NPM(K)  PM(K)  WS(K) VM(M)  CPU(s)  Id ProcessName
-----  -
          318     25   36584   52924   164    14.95   2156 firefox
```

```
PS C:\Documents and Settings\razvan> ps power*
Handles  NPM(K)  PM(K)  WS(K) VM(M)  CPU(s)  Id ProcessName
-----  -
          674     70  110888    5036    315   391.47   1500 POWERPNT
          558     15    68028   63388    571    4.64    496 powershell
```

```
PS C:\Documents and Settings\razvan> get-item -path alias:ps
CommandType Name Definition
-----
Alias ps Get-Process
```

- Comanda ps (process status)
 - informații despre procesele care rulează pe sistem la un moment dat

```
razvan@anaconda:~$ ps
  PID TTY          TIME CMD
 10388 pts/2    00:00:00 bash
 10478 pts/2    00:00:00 ps
razvan@anaconda:~$ ps -e
  PID TTY          TIME CMD
    1 ?            00:00:00 init
    2 ?            00:00:00 ksoftirqd/0
    3 ?            00:00:00 events/0
    4 ?            00:00:00 khelper
    5 ?            00:00:00 kacpid
...
 3798 ?            00:00:08 soffice.bin
 10388 pts/2    00:00:00 bash
 10415 ?            00:00:00 sshd
 10479 pts/2    00:00:00 ps
```

```

razvan@anaconda:~$ ps -o pid,ppid,TTY,rss,cmd -e --sort rss
  PID  PPID  TT      RSS  CMD
 3263     1  ?      380  /usr/sbin/squid -D -sYC
 2770  2746  ?      388  hald-addon-storage: polling /dev/hdc
 2737     1  ?      392  /usr/bin/dbus-daemon --system
 1075     1  ?      400  udevd --daemon
 3355     1  tty1    416  /sbin/getty 38400 tty1
 3356     1  tty2    416  /sbin/getty 38400 tty2
[...]
 3364     1  tty6    416  /sbin/getty 38400 tty6
 3057     1  ?      432  /usr/bin/xfstools -daemon
 3380  3377  ?      436  git-daemon --verbose --base-path=/var/cache /var/cach
 2733     1  ?      448  /usr/sbin/courierlogger courierpop3login
 3636     1  ?      456  gpg-agent -daemon
[...]
   367  3313  ?      5188 /usr/sbin/apache
 2827  2804  ?      5336 /usr/bin/python /var/lib/mailman/bin/qrunner --runner
[...]
 2823  2804  ?      6652 /usr/bin/python /var/lib/mailman/bin/qrunner --runner
 32488 3313  ?     10340 /usr/sbin/apache
 3313     1  ?     10488 /usr/sbin/apache

```

```
razvan@anaconda:~$ ps -H -o pid,ppid,comm -C init,sshd,bash,ps
```

```

PID  PPID  COMMAND
  1      0  init
3048      1  sshd
29263 3048  sshd
29266 29263  sshd
29267 29266  bash
32688 3048  sshd
32691 32688  sshd
32692 32691  bash
32733 3048  sshd
32736 32733  sshd
32737 32736  bash
  487 32737  ps

```

```
razvan@anaconda:~$ pstree -p
```

```

init(1)---Xprt(3144)
      |---apache(3313)---apache(27892)
      |                   |---apache(28002)
      |                   |---apache(28037)
      |                   |---apache(28038)
[... ]
|---sshd(3048)---sshd(32733)---sshd(32736)---bash(32737)---pstree(571)

```

- Vizualizarea dinamică (în timp real) a funcționării sistemului
 - memoria ocupată
 - ocuparea procesorului
 - informații despre procese

Utilitarul top – processes gone crazy

```

razvan@asgard: ~
File Edit View Terminal Tabs Help
top - 11:53:29 up 9 days, 2:57, 1 user, load average: 13.18, 13.23, 13.18
Tasks: 84 total, 14 running, 70 sleeping, 0 stopped, 0 zombie
Cpu(s): 90.4% us, 9.6% sy, 0.0% ni, 0.0% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 256740k total, 253688k used, 3052k free, 11468k buffers
Swap: 289128k total, 288812k used, 316k free, 63352k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 27840 www-data  25   0 29428 5360  12m  R  10.3   2.1 266:50.19 apache
 30693 www-data  25   0 24808 5400  12m  R  10.0   2.1 229:53.78 apache
 22955 www-data  25   0 27408 4912  12m  R   9.0   1.9 527:42.72 apache
  9899 www-data  25   0 27628  11m  12m  R   9.0   4.5  26:47.05 apache
 26351 www-data  25   0 27852 4320  12m  R   7.0   1.7  3414:09 apache
  6404 www-data  25   0 31420 4332  12m  R   7.0   1.7  2696:37 apache
 13351 www-data  25   0 27712 4900  12m  R   7.0   1.9  1030:09 apache
 17978 www-data  25   0 28964 4700  12m  R   7.0   1.8  773:08.76 apache
 11244 www-data  25   0 28132 4932  12m  R   7.0   1.9  576:26.14 apache
 10190 www-data  25   0 27704  12m  12m  R   7.0   5.1  14:45.01 apache
  6097 www-data  25   0 24612 4860  12m  R   6.6   1.9  1096:31 apache
 24367 www-data  25   0 27796 4884  12m  R   6.6   1.9  797:24.73 apache
 30071 www-data  25   0 32868 4784  12m  R   6.6   1.9  636:45.04 apache
     1 root      16   0  1504  452 1352  S   0.0   0.2   0:03.71 init
     2 root      34  19     0     0     0  S   0.0   0.0   0:00.03 ksoftirqd/0
     3 root       5 -10     0     0     0  S   0.0   0.0   0:00.00 events/0
     4 root      11 -10     0     0     0  S   0.0   0.0   0:00.00 khelper
  
```

- Executabil (program)
 - fișier al cărui conținut este rulat de SO
 - conține reprezentarea instrucțiunilor unui procesor (cod mașină)
- Mai multe secțiuni
 - date inițializate
 - date neinițializate
 - cod (instrucțiuni)
- Fișierele obiect conțin aceleași informații (secțiuni)
- Un fișier executabil este obținut prin legarea (linking) mai multor fișiere obiect

- Proces
 - imaginea executabilului care rulează
 - alte informații necesare SO
- Date dinamice
 - heap – pentru alocare dinamica (malloc)
 - stivă – alocarea variabilelor locale funcțiilor
- Informații necesare planificatorului
 - prioritate
 - cantă de timp
 - identificator (PID)
- Alte proprietăți: fișiere deschise, semnale, limitări memorie

- Sistem de fişiere virtual
 - informațiile se găsesc în memorie
- Montat în /proc
- Folosit de multe utilitare de bază

```
razvan@anaconda:~$ dpkg -L procps  
/bin/kill  
/bin/ps  
[...]  
/usr/bin/uptime  
/usr/bin/tload  
/usr/bin/free  
/usr/bin/top  
/usr/bin/vmstat  
/usr/bin/watch  
/usr/bin/skill  
/usr/bin/pmap  
/usr/bin/pgrep  
/usr/bin/slabtop
```

- Informațiile despre un proces sunt grupate într-un director dat de pid-ul procesului

```

anaconda:~# ls /proc/
1          2693  2770  3          32398  3377  853          driver      modules
10         2694  2798  3015     32488  3379  854          execdomains mounts:

```

- Informații despre un proces (vim)

```

anaconda:~# ps -e | grep vim
  877 pts/1    00:00:00 vim
anaconda:~# cd /proc/877/
anaconda:/proc/877# ls
attr      cpuset   exe      mem      oom_adj  smaps    status
auxv      cwd      fd       mounts  oom_score stat     task
cmdline   environ maps     mountstats root      statm    wchan
anaconda:/proc/877# ls -l exe
lrwxrwxrwx 1 root root 0 Oct 20 11:33 exe -> /usr/bin/vim.basic
anaconda:/proc/877# cat cmdline
vimmac_add.txtanaconda:/proc/877# ls fd/
0 1 2 4
anaconda:/proc/877# ls -l fd/
total 4
lrwx----- 1 root root 64 Oct 20 11:34 0 -> /dev/pts/1
lrwx----- 1 root root 64 Oct 20 11:34 1 -> /dev/pts/1
lrwx----- 1 root root 64 Oct 20 11:33 2 -> /dev/pts/1
lrwx----- 1 root root 64 Oct 20 11:34 4 -> /root/.mac_add.txt.swp

```

- Un proces este creat din cadrul unui proces existent
- În Unix, init este părintele tuturor proceselor
- Fazele creării
 - copierea procesului părinte (cu imaginea acestuia)
 - alocarea resurselor necesare pentru noul proces (pid, tabelă de fișiere, zone de memorie etc.)
 - încărcarea executabilului asociat (imaginea procesului)
 - actualizarea resurselor (zone de memorie, fișiere deschise)
 - ulterior se planifică procesului

- La terminarea unui proces
 - toate informațiile asociate dispar
 - resursele (memorie, fișiere) sunt eliberate
- Când un proces moare, se șterge fișierul executabil asociat?
- Când moare un proces?
 - se ajunge la sfârșitul zonei de cod a executabilului
 - se apelează **exit** (sau o funcție asemănătoare)
 - condiții anormale (abnormal termination): primirea unui semnal

- Mecanisme prin care se indică unui proces
 - încheierea unei operații sau atingerea unei stări (notificare)
 - o condiție deosebită în funcționare (ex. accesarea unei zone invalide de memorie)
- Mecanismele echivalente într-un sistem Windows
 - events, pentru notificare
 - excepții, pentru condiții deosebite

- Semnalele Unix

```
razvan@anaconda:~$ kill -l
```

```

1) SIGHUP      2) SIGINT     3) SIGQUIT    4) SIGILL
5) SIGTRAP    6) SIGABRT   7) SIGBUS     8) SIGFPE
9) SIGKILL    10) SIGUSR1  11) SIGSEGV   12) SIGUSR2
13) SIGPIPE   14) SIGALRM  15) SIGTERM   17) SIGCHLD
18) SIGCONT   19) SIGSTOP  20) SIGTSTP   21) SIGTTIN
22) SIGTTOU   23) SIGURG   24) SIGXCPU   25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF  28) SIGWINCH  29) SIGIO
30) SIGPWR    31) SIGSYS

```

- Comanda **kill** poate fi folosită pentru transmiterea unui semnal
 - denumire nefericită (nu toate semnalele termină procesul)

```

razvan@anaconda:~$ ps
  PID TTY          TIME CMD
 10388 pts/2    00:00:00 bash
 10422 pts/2    00:00:00 netcat
 10424 pts/2    00:00:00 ps
razvan@anaconda:~$ kill -KILL 10422
[1]+  Killed                  netcat
razvan@anaconda:~$ ps
  PID TTY          TIME CMD
 10388 pts/2    00:00:00 bash
 10425 pts/2    00:00:00 ps
razvan@anaconda:~$

```

- Combinații de taste în shell pentru transmiterea de semnale
 - CTRL-C – transmite SIGINT (întrerupe procesul – terminare)
 - CTRL-\ - transmite SIGQUIT (oprește procesul – terminare – mai puternic decât CTRL-C)
 - CTRL-Z – transmite SIGSTP (oprește procesul – pauză)

- “Social” processes 😊
- O primă formă de comunicare sunt semnalele
 - un proces poate transmite un semnal altuia
 - comanda kill – transmiterea unui semnal de la shell la proces
- Alte forme de comunicație între procese
 - **cozi de mesaje**
 - sunt folosite ‘căsuțe de mesaje’
 - **memorie partajată**
 - zone de memorie partajate între mai multe procese
 - Comunicație facilă (scriere/citire în/din memorie)
 - **socketi**
 - comunicația prin rețele de calculatoare

- Formă de bază de comunicare între procese
 - ieșirea unei comenzi este **redirectată** spre intrarea celei de-a doua comenzi

```
anaconda:~# ps -e | grep bash
```

```
32737 pts/1      00:00:00 bash
```

```
794 ?           00:00:00 bash
```

```
854 pts/1      00:00:00 bash
```

```
anaconda:~# ps -e | wc -l
```

```
118
```

```
anaconda:~# ps -e | head -n 5
```

PID	TTY	TIME	CMD
1	?	00:00:02	init
2	?	00:00:00	migration/0
3	?	00:00:00	ksoftirqd/0
4	?	00:00:00	events/0

- Fundal (background) sau prim plan (foreground)
- Rulare în background – nu se pot primi comenzi de la utilizator
 - se folosește operatorul **&**
 - procesul continuă să ruleze

```

anaconda:/# du -hs &
[1] 1169
anaconda:/# ps -e | grep du
 1169 pts/1      00:00:00 du
anaconda:/# top -b -n 1 | grep du
 1169 root      18   0  1624   564   356 R   0.0   0.2   0:00.80 du
anaconda:/# fg
du -hs

```

- Comanda **fg** aduce procesul în foreground
- Folosirea CTRL-Z (SIGSTP) forțează procesul în background (fără a rula)

- Disk And Execution MONitor (backronym)
- Proces care rulează în background
 - detașat de terminalul de control
 - orice daemon are ca părinte init
- Numele se termină, de obicei, în d (sshd, syslogd, httpd, etc.)
- Cum se interacționează cu un daemon?
 - semnale
 - fișiere de configurare

- Pe Windows, echivalentul daemon-ului este serviciul: **Control Panel -> Administrative Tools -> Services**
- Intreacțiune cu daemon-ii în Debian/Ubuntu – scripturile din /etc/init.d

```

anaconda:/# /etc/init.d/apache stop
Stopping apache 1.3 web server....
anaconda:/# /etc/init.d/apache start
Starting apache 1.3 web server....
anaconda:/# /etc/init.d/apache restart
Restarting apache 1.3 web server....

```

- În Windows, comanda **net** sau interfața grafică

```

C:\>net stop "DHCP client"
The DHCP Client service is stopping.
The DHCP Client service was stopped successfully.

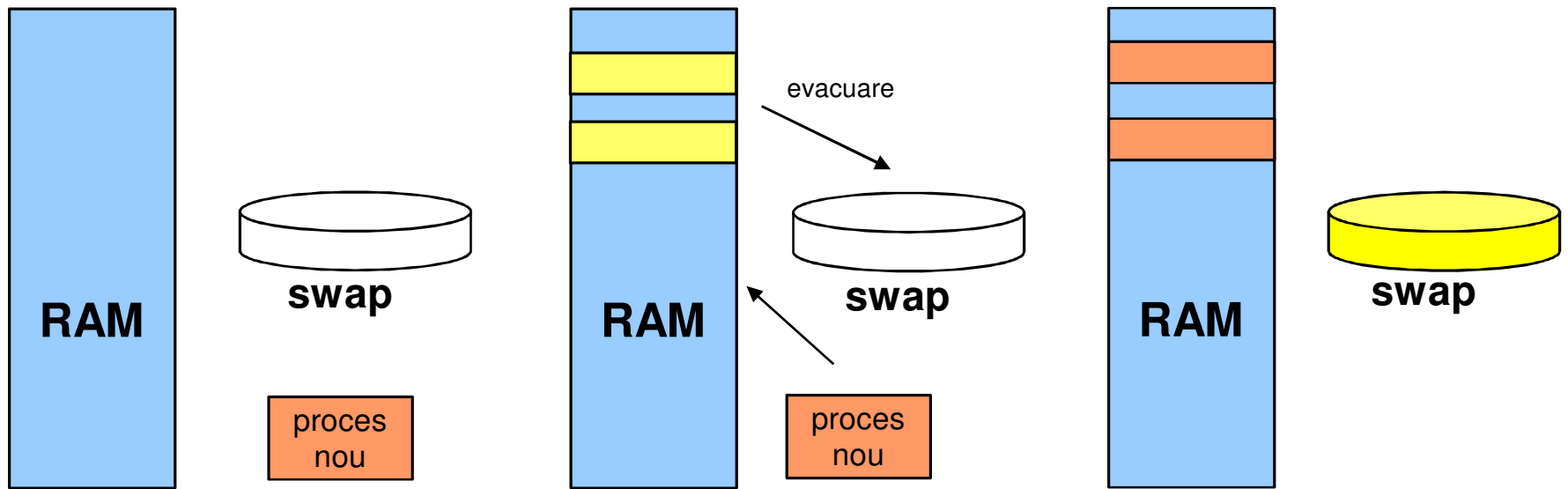
```

```

C:\>net start "DHCP client"
The DHCP Client service is starting.
The DHCP Client service was started successfully.

```

- Ce se întâmplă când memoria este insuficientă (prea multe procese)?
 - transfer de pagini între memorie și disc – swap
- În Windows se folosește partiția de sistem
- În Linux, se obișnuiește prezența unei partiții de swap dedicate
 - # **mkswap /dev/sda2**
 - se poate folosi și un fișier
 - # **mkswap file.swap**



- proces
- program/executabil
- context de execuție
- schimbare de context
- context switch
- cantă de timp
- multitasking
- multiuser
- planificator de proces
- pid
- Init
- ierarhie de procese
- ps
- Task manager
- Get-Process (PowerShell)
- top
- comunicație între procese
- semnale
- kill
- foreground
- background
- daemon
- serviciu
- swapping

- Cărți

- Silberschatz, Galvin, Gagne – Operating System Concepts, 7th Edition (chapter 3: Processes)
- Andrew Tanenbaum – Modern Operating Systems, 2nd Edition (chapter 2: Processes and Threads)

- Link-uri

- [http://en.wikipedia.org/wiki/Process_\(computing\)](http://en.wikipedia.org/wiki/Process_(computing))
- [http://en.wikipedia.org/wiki/Daemon_\(computer_software\)](http://en.wikipedia.org/wiki/Daemon_(computer_software))
- http://en.wikipedia.org/wiki/Category:Unix_signals
- http://en.wikipedia.org/wiki/Inter-process_communication
- <http://en.wikipedia.org/wiki/Paging>
- <http://computer.howstuffworks.com/operating-system5.htm>

?

