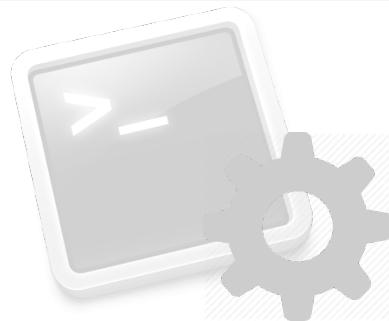


Resource Management



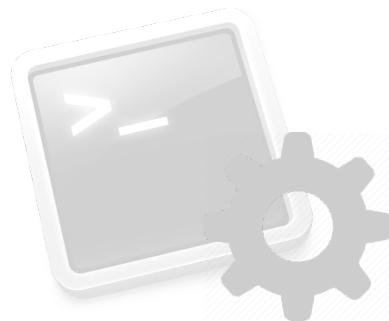
Gestión de Recursos

1



Index (Resources managed)

- **CPU**
 - Monitoring.
 - Control.
 - Scheduling services.
- **Memory**
 - Monitoring.
 - Paging.
- **Disk**
 - Monitoring.
 - Quotas.



Gestión de Recursos

2



CPU

- **CPU Management:** Critical aspect for correct system utilization.

- Monitor and supervise system status.
- Manage process priorities, favoring those requiring more computing resources.
- Make use of CPU resources during low utilization hours.



Gestión de Recursos

3



CPU

```
calderon:~> vmstat 5 3
procs      -----memory-----  ---swap--  -----io----  --system--  ----cpu----
 r b    swpd   free   buff  cache    si   so   bi   bo   in   cs   us   sy   id   wa
 0 1 132908 2106268 22712 1609316    0   0   15    1   11   17   3   1 92   4
 0 1 132908 2106028 22832 1609468    0   0   0 1254 1730   983   0   1 80 19
 0 1 132908 2105980 22952 1609620    0   0   2 302 1599   553   0   1 84 15
```

- **CPU Monitoring:**

- Command **vmstat**: global instantaneous utilization.
 - Syntax: #vmstat <interval> <samples>
 - Equivalent command for multiprocessors: mpstat
 - Monitoring and supervising system status.
 - (r/b): processes running/waiting or sleeping.
 - us: % of user time (including nice time), execution of code not belonging to the kernel
 - sy: % of time executing kernel code.
 - id/wa: % of idle or input/output time.

- Command **uptime**: global average utilization.

- Reports the following information: Current time, how long running, users logged on and system load (Average load values in 1, 5 and 15 minutes).
- Load: number of processes in runnable (running or waiting to run) or I/O Waiting state
 - Not a normalized value, Load=1 does not mean the same for 1CPU or 4CPU systems.

```
calderon:~> uptime
17:39:03 up 17 days,  9:23,  2 users,  load average: 0,06, 0,06, 0,05
```

Gestión de Recursos

4



CPU

• CPU Monitoring:

```
alumno@dasovm: /usr/src/linux-source-2.6.18/drivers/net
File Edit View Terminal Tabs Help
top - 10:18:58 up 5:20, 4 users, load average: 0.22, 0.18, 0.07
Tasks: 91 total, 2 running, 89 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.0%us, 0.7%sy, 0.0%ni, 98.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 510872k total, 482148k used, 347744k free, 21804k buffers
Swap: 40960k total, 12k used, 409604k free, 220104k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3859	alumno	15	0	62464	15m	11m	S	0.0	3.1	0:02.53	nautlius
2902	root	15	0	39592	14m	6300	S	1.0	2.9	0:38.93	Xorg
3857	alumno	15	0	23520	14m	9.8m	S	0.0	2.9	0:03.40	gnome-panel
4018	alumno	15	0	31976	12m	9136	R	0.0	2.5	0:01.46	gnome-terminal
3867	alumno	15	0	21936	12m	9260	S	0.0	2.5	0:00.78	wnck-applet
3882	alumno	15	0	21456	12m	9384	S	0.0	2.4	0:00.70	update-notifier
3984	alumno	15	0	22356	11m	8972	S	0.0	2.3	0:00.50	mixer-applet2
3893	alumno	15	0	29568	11m	8988	S	0.0	2.3	0:00.52	applet
3770	alumno	15	0	21156	11m	8988	S	0.0	2.2	0:00.52	gnome-session-manag
3966	alumno	15	0	23568	10m	8380	S	0.0	2.1	0:00.36	clock-applet
5097	alumno	15	0	17716	9860	6004	S	0.0	1.9	0:00.28	notification-da
3846	alumno	15	0	16388	9672	7196	S	0.0	1.9	0:01.17	metacity
3843	alumno	15	0	28916	9256	7264	S	0.0	1.8	0:00.65	gnome-settings
3910	alumno	15	0	54272	7736	6352	S	0.0	1.5	0:08.77	gnome-cups-icon
3982	alumno	15	0	16948	7244	6640	S	0.0	1.4	0:00.19	notification-ar
3999	alumno	15	0	16508	6320	5164	S	0.0	1.2	0:00.53	gnome-screensav

- Command **top**: global/individual dynamic utilization

- Allows to know system load in real time.
- Provides much more detailed information
- Interactive “shell”, can kill processes, modify priorities, etc. (h for extended help).

- Command **ps**: Individual utilization

- top makes use of this command to show its results.

* All these commands make use of /proc/ (kernel state tables)

CPU

```
calderon:~> sa
579      222.81re    0.16cp    7220k
4        0.36re    0.12cp   31156k  up2date
8        0.02re    0.02cp   16976k  rmpmq
8        0.01re    0.01cp   2148k   netstat
11       0.04re    0.00cp   8463k   grep
```

• Accounting: Service to track system usability.

- Useful when systems were expensive (pay per CPU cycles).
- Currently: analysis of utilization profiles
 - For example, to plan the purchase of new equipment.
- **Login accounting**: accounts for system accesses.
 - /var/log/wtmp updated by init after every system login (no readable file).
 - The command **last** allows to see the content of wtmp file.
 - The command **ac** displays statistics about how long users have been logged on.
- **Process accounting**: accounts for system utilization (processes)
 - Command **lastcomm**: lists the last command executed by any user, from any terminal
 - Command **sa**: summarize information about executed commands (cpu, io, mem)
- **acct** package: ac + lastcomm + sa.

CPU

• Process Control:

- Command **kill**: process manipulation (much more than killing).

- Syntax: #kill <options> PID

- Option -STOP: stop the process (-CONT to rerun the process)
 - Option -9: Kill the process (killall -9 user)
 - kill as a combined command: # ps -ef | grep user | awk '{print \$1}' xargs kill -STOP

- Process priority: scheduler of multi-process systems.

- The scheduler allocates time intervals according to priority (column PR in top)
 - Users can “partially” regulate priority (column NI in top)
 - From -20 (max priority) to 19 (min priority)
 - If it is not modified externally, a process inherits its father priority.

- Command **nice**: Change the inherited priority of a process.

- Syntax: nice -n +value command (+ decreases priority, - increases).
 - **renice** allows to change command (or group of commands) priority during execution.
 - renice +-value [-p PID] [-u user] [-g group]

PR = 20 + NI

CPU

• Process Control:

- Command **ulimit**: limit the utilization of system resources.

- Limits established to the shell we are working in

- Syntax: #ulimit -<option> [limit]

- Option -a: informs about the current limits of all resources.
 - Option -f: max number of files created by the shell
 - Option -m: max available memory.
 - Option -t: max amount of CPU time (seconds)
 - ...

```
calderon:~> ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
file size               (blocks, -f) unlimited
max locked memory       (kbytes, -l) unlimited
max memory size         (kbytes, -m) unlimited
open files              (-n) 256
pipe size                (512 bytes, -p) 1
stack size              (kbytes, -s) 8192
cpu time                 (seconds, -t) unlimited
max user processes      (-u) 709
virtual memory           (kbytes, -v) unlimited
```

CPU

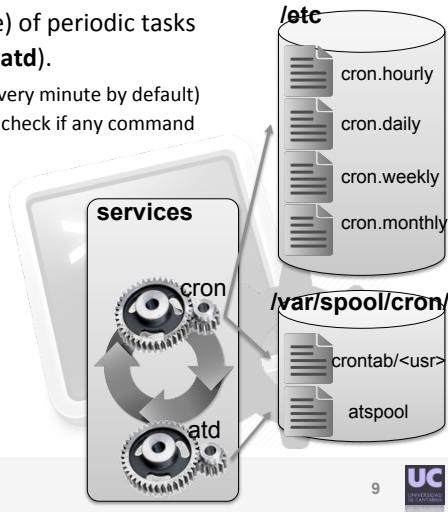
- **Scheduling Services**

- Programmed execution (in the future) of periodic tasks (daily, weekly, ...)(**cron**) or one-time (**atd**).

- cron and atd services read periodically (every minute by default) the content of their configuration files to check if any command must be executed.

- Some examples of periodic tasks:

- Log rotation (next chapter)
- Deletion of /tmp directory
- Backups
- Database update(man, locate, etc.)



Gestión de Recursos

9

CPU

- **Scheduling Services**

- File **crontab**: one line per programmed task.

- crontab -l: list all the tasks programmed
- crontab -r: delete programmed tasks
- crontab -e: edit file crontab.
- Examples:
 - * * * * * <command>
 - 9 0 1-7,9-16 * * 1-5 <command>



- **Security:**

- /etc/cron.allow /etc/cron.deny
- /etc/at.allow /etc/at.deny

- Commando **at**: daemon to control atd.

- Sending a task: at TIME (it opens its own shell, where commands are specified)
 - #> at 13:00
 - at> ls -R /
- Standard output via mail (can be redirected to a file)
- See pending tasks: at -l
- Remove tasks: at -d <job>

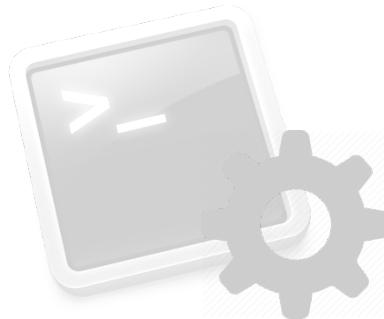
Gestión de Recursos

10



Index (Resources managed)

- **CPU**
 - Monitoring.
 - Control.
 - Timing services.
- **Memory**
 - Monitoring.
 - Paging.
- **Disk**
 - Monitoring.
 - Quotas.



Memory

- **Memory Monitoring:**
 - Command **vmstat**: Global instantaneous utilization.
 - Syntax: #vmstat <interval> <samples>
 - Equivalent command for multiprocessors: mpstat
 - Monitoring and supervising system status.
 - (swpd): Amount of virtual memory in use.
 - (free): Amount of free memory.
 - (buff): Amount of memory employed as buffer
 - (cache): Amount of memory employed as cache.

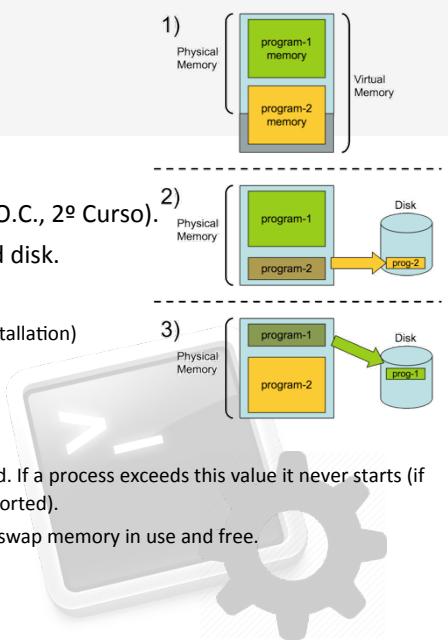


```
calderon:~> vmstat 5 3
procs --memory-- swap-- io-- system-- cpu--
 r b swpd free buff cache si so bi bo in cs us sy id wa
 0 1 132908 2106268 22712 1609316 0 0 15 1 11 17 3 1 92 4
 0 1 132908 2106028 22832 1609468 0 0 0 1254 1730 983 0 1 80 19
 0 1 132908 2105980 22952 1609620 0 0 2 302 1599 553 0 1 84 15
```

Memory

• Swapping

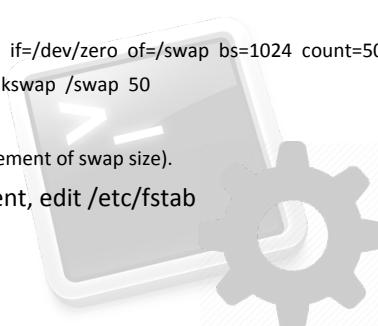
- Virtual Memory > Physical Memory (O.C., 2º Curso).
- Page exchange between memory and disk.
- Swapping space (swap)
 - At least one partition required (during installation)
 - It appears in /etc/fstab
 - Swap size depends on system utilization
 - Workstation: SIZE-swap = SIZE-mem
 - Server: SIZE-swap = 2*(SIZE-mem)
 - This parameter must be carefully analyzed. If a process exceeds this value it never starts (if it is exceeded dynamically execution is aborted).
 - The command **free** shows the amount of swap memory in use and free.



Memory

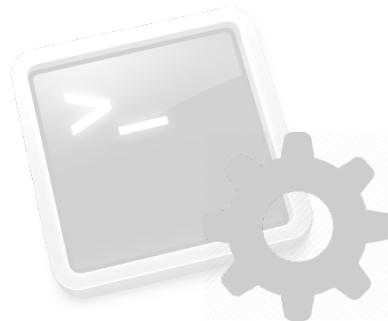
• Modification of Swap size

- The amount of memory employed for swapping can be dynamically increased.
 - The swap partition is hard to modify, the alternative consists on adding special swap files.
- Creating a swap file:
 - Create an empty file of 50 block size: # dd if=/dev/zero of=/swap bs=1024 count=50
 - Mark it to be identified by the kernel: # mkswap /swap 50
- Activation of the file as swap space:
 - #swapon /swap (verify with free the increment of swap size).
- If we want the change to be permanent, edit /etc/fstab
 - /swap swap swap defaults 0 0



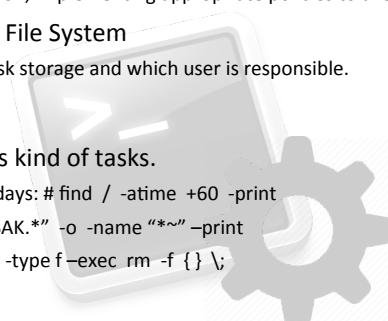
Index (Resources managed)

- **CPU**
 - Monitoring.
 - Control.
 - Timing services.
- **Memory**
 - Monitoring.
 - Paging.
- **Disk**
 - Monitoring.
 - Quotas.



Disk

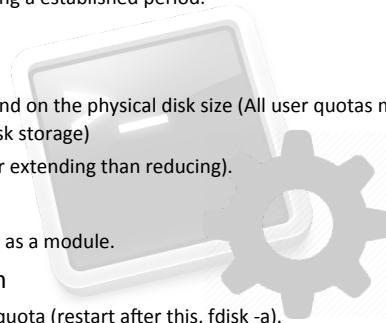
- **Disk Monitoring:**
 - Command **df**: % of occupation of the mounted file systems.
 - Syntax: #df -<options>
 - Option -h: Human readable sizes.
 - We can prevent File System from saturation, implementing appropriate policies to avoid it.
 - Command **du**: size of a branch in the File System
 - Allows us to know where is the leak of disk storage and which user is responsible.
- **Searching for “trash” content**
 - find is a very useful command for this kind of tasks.
 - Look for files not used during the last 60 days: # find / -atime +60 -print
 - Look for backup files: # find / -name ".BAK.*" -o -name "*~" -print
 - Automatic removal: # find / -atime +60 -type f -exec rm -f {} \;



Disk

- **Disk Quotas:**

- Allows to control the amount of data and files for each user/group in the FS.
 - Delegated disk management, users must control their info or they consume their quota.
- Quota system defines two different limits:
 - Soft limit: can be exceeded, but only during a established period.
 - Hard limit: can never be exceeded.
- How to set those limits?
 - Depends on the user (tasks performed) and on the physical disk size (All user quotas must never surpass the amount of available disk storage)
 - Recommended to be conservative (better extending than reducing).
- Requires quota support in the kernel
 - Usually, every kernel sets it up by default as a module.
- Requires setting up in the File System
 - Modify /etc/fstab with usrquota and grpquota (restart after this, fdisk -a).



Disk

- **Disk Quotas:**

- Command **edquota**: modify the limits of a user/group.
 - Syntax: edquota -<options> [user] [-g group]
 - Starts a text editor for limit modification
 - Limits are modified in 1Kbyte blocks and inodes (Null values mean unlimited quota).
 - Option -p: copy quota values between users (# edquota -p user1 user2)
 - option -t: change the pardon period of the soft quota.
- Command **quotaon/quotaoff**: power on/off quotas system.
- Command **quotacheck**: verify the integrity of quota system.
- Command **repquota**: reports the content of quota system database
 - Files quota.user and quota.grp
- quota -v user: see quota and status of a user.

