

Advanced Linux System Administration

Topic 8. Resource management



Pablo Abad Fidalgo
José Ángel Herrero Velasco

Departamento de Ingeniería Informática y Electrónica

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Index

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

CPU

- 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 (**scheduling**).

CPU Monitoring

- Command **vmstat** (mpstat): global instantaneous utilization:

- Syntax: **#vmstat <interval> <samples>**:
 - (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.

```
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	32908	2106268	22712	1609316	0	0	15	1	11	17	3	1	92	4
0	1	32908	2106028	22832	1609468	0	0	0	1254	1730	983	0	1	80	19
0	1	32908	2105980	22952	1609620	0	0	2	302	1599	553	0	1	84	15

- 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
```

CPU Monitoring

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

- Command **top**: global/individual dynamic utilization:
 - Allows knowing 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.
- **Process accounting**: accounts for system utilization (processes):
 - Command **lastcomm**: lists the last command executed by any user, from any terminal.
 - Command **sa**: summarizes information about executed commands (cpu, io, mem).
 - **acct** package: ac (login accounting) + lastcomm + sa.

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

CPU Management

- **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: **PR = 20 + NI**

- 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]`

CPU Management

- **Process Control:**

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

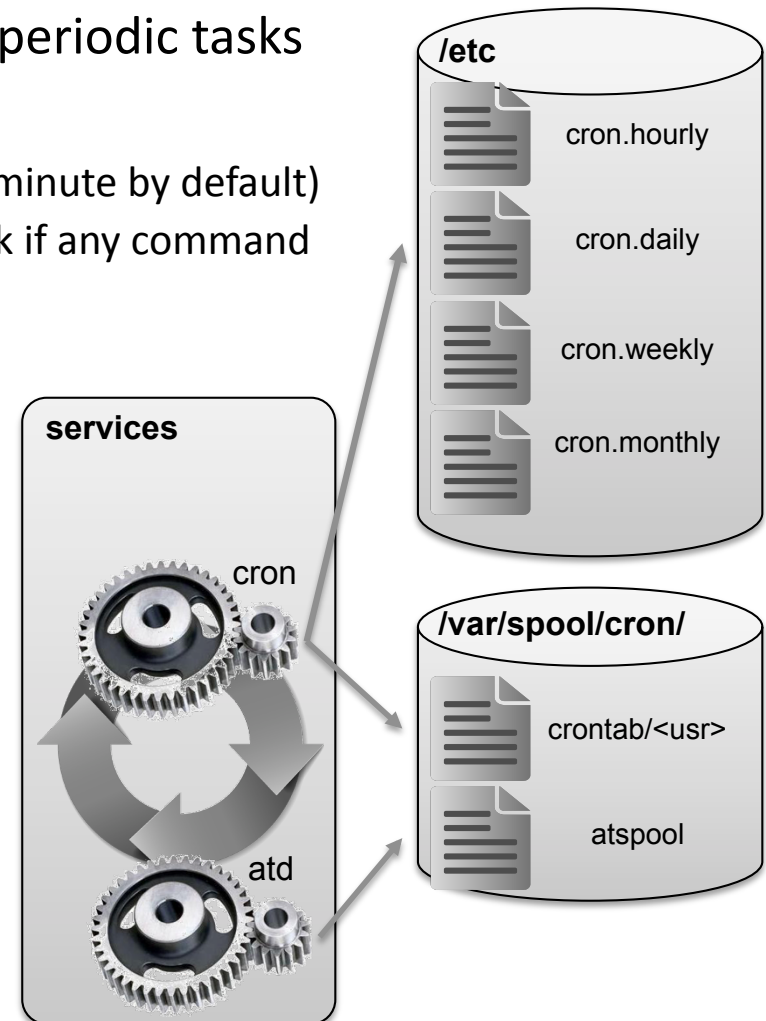
- Limits established to the shell we are working in.
 - Syntax: `#ulimit -<option> [limit]`:
 - Option `-a`: informs about the current limits of all resources.
 - Option `-f`: max number of files created by the shell.
 - Option `-m`: max available memory.
 - Option `-t`: max amount of CPU time (seconds).
 - ...

```
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

- **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 section).
 - Deletion of /tmp directory.
 - Backups.
 - Database update (man, locate, etc.).



CPU Scheduling

- **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>.

- 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>.



- **Security:**

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

Index (Resources managed)

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

Memory Management

- **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.

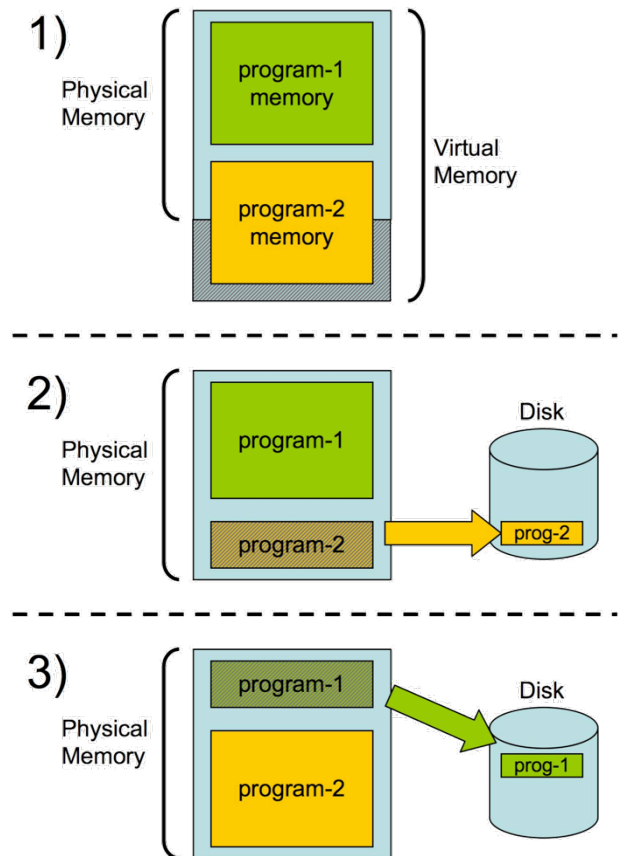
```
calderon:~> vmstat 5 3
```

| 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 Management

- **Swapping:**

- Virtual Memory > Physical Memory (O.C., 2nd Course).
- 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: $\text{SIZE-swap} = \text{SIZE-mem}$.
 - Server: $\text{SIZE-swap} = 2 * (\text{SIZE-mem})$.
 - 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 Management

- **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 of 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 Management:
 - Monitoring.
 - Control.
 - Timing services.
- Memory Management:
 - Monitoring.
 - Paging.
- **Disk Management:**
 - Monitoring.
 - Quotas.

Disk Management

- **Disk Monitoring:**

- Command **df**: % space available on the mounted File Systems:
 - Syntax: `#df -<options>`:
 - Option `-h`: Human readable sizes.
 - We can prevent File System from saturation, by implementing appropriate policies to avoid it.
- Command **du**: size of a branch in the File System:
 - Allows us to know where the leak of disk storage is and which user is responsible.

Disk

- **Disk Quotas:**
 - Allows controlling 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 are those limits set?:
 - Depends on the user (tasks performed) and on the physical disk size (all user quotas must never surpass the amount of available disk storage).
 - Advisable to be conservative (better to extend than to reduce).
 - 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.grpb`.
- `quota -v user`: see quota and status of a user.