41 - Programs Processes and Scripts

Processes commands in Linux

- **Types of processes**
  - Programs
  - Daemons (detach themselves from the process that launched it)
- **Commands to display the processes**

ASCII Programs

- `ps cauxOt | less`
  `c`=Commands only without parameters `a`=all started from tty
  `u`=process owner (user) `x`=All Processes not started from tty
  `Ot`=Order of process types `| less`=show scrollable output
- `ps fC httpd` Shows all the *httpd* processes
- `ps caux | grep httpd` " " " " "
- `ps cauxf | less` Shows processes with hierarchy tree
- `pstree p | less` (processes hierarchy tree with PID)
- `ps -eo %p%P%U%n%a`
  Custom list -> PID-PPID,User-Nice-Cmd+Args
- `ps -eo %p%u%n%y%x%c%a --sort user | less`
  Custom list -> PID-User-Nice-tty-Time-Command-CmdArgs
- `ps -eo pid,ppid,user,nice,comm,args --sort user | less`
  Custom list -> PID-PPID-User-Nice-Command-CmdArgs
- `ps -eo pid,user,bsdstart,comm,args | less`
  Custom list ->PID-User-StartTime-Command-CmdArgs
- `pidof` *`processname`* Gives a list of PID's for this process
- `top` Curent processes with refreshes.
  Type 'h' for help screen
- `watch -n 1 ps caux` Refreshes every second
- `alias wps="watch -n1 'ps caux | grep` *`xxxx`* `'"`

X-Programs

- `kpm` is standard
- `kpstree` (Good)
- `ksysguard` from SuSE CD (Good)
- `gtop` and `procman` from the `gnome` packages

- **Ending processes**

ASCII Programs

`kill` <*ProzessNummer*> (Default is `-15` SIGTERM)
`kill -9` <*ProzessNummer*>
`killall` <*Prozessname*> (Default is `-15` SIGTERM)
`killall -9` <*Prozessname*>
`kill -l` Lists the kill messages options
When a process dies it sends a **SIGHUP** to all of its children

X-Programs

- `kpm` and `ksysguard` pakages are standard (Good)
- `ktop` on CD only with older Distributions as 8.1
- `xzap` on CD only with older Distributions as 8.1
- `qps` on CD

- **Show lots of info about the system processes and kernel setup etc.**
  `procinfo [-afn1]`
  `-f` refreshes (default every 5 sec)
  `-a` shows all info possible
  `-n1` refreshes every sec

- **Starting a program with another Nice value:**
  `nice [-n priority] ProgramName`
  Note 1: `nice` needs a first '–' before the priority. or `-n`
  `eg. nice --12 prgm` (starts `prgm` with nice value of `-12`)
  or `nice -n -12 prgm` (does the same)
  Note 2 : Default nice values:
  - Starting program without nice: `0`
  - Starting program with nice: `+10`

- **To show all the modules or library modules (`.so`) a program uses:**
  `ldd ProgramName`

- **Changing the NICE priority for an already running process:**
  `renice priority PID`
  priority is : `-20` to `19`.
  Note: Users cannot assign a new priority higher that the current one

- **Sarting a new process after the last background process has finished**
  `wait $! ; NewProcess`

- **Processes PIDs:**

`/var/run/`	Where some PID of running processes appear
`pidof processname`	List the PID's of the running process name
`/bin/kill -p processname`	List the PID's of the running process name

- **Role of `/proc` directory : Stores a directory per process**

- **`ps` Status column Description**

  `D` – Uninterruptible sleep
  `R` – Runnable (On the running queue)
  `S` – Sleeping
  `T` – Traced or Stopped
  `Z` – Zombie ....dead but not completely taken out of system yet.
  `W` – No Resident Pages reserved.
  `X` – Dead process(should never be seen)

  BSD format extra status
  `<` – Niced to Higher priority as 0 (-1 to -20)
  `N` – Niced to lower priority as 0(1 to 19)
  `L` – Pages locked in to memory (Custom IO -Realtime system)
  `s` – Session leader
  `l` – Multithreaded
  `+` – Foreground process

- **Display of system memory usage:**

  ASCII Programs

  ```
  free
  cat /proc/meminfo
  top, htop
  ```

  X-Programs

  ```
  xosview, ksysguard
  ```

- **To Kill all processes that use a mounted filesystem to free it up before unmounting it:**
  `fuser -km mountpoint`

- **Bash Jobs**

Command &	Start an active job in background Shows also job number in square brackets and PID
%[+-n] or fg	Activates the last job in background [+-] and put it in foreground.
fg %*jobNr* or %*jobNr*	Activates a job and puts it in the foreground
or jobs -x *Programname*	
bg	Activates the last job and put it in background.
bg %*jobNo.* or %*jobNo.* &	Activates a stopped job and keeps it in the Background
<Ctrl>-Z	Sends foreground job as stopped in background
%[+-n]& or bg [+-n]	Resumes the last stopped job as running in the background
jobs	List all the jobs of this shell The result will be shown as follows: (*x* is the job number) [*x*]+ The last job started in the background [*x*]- The job started in the background before the last one.