

GNU/Linux

Processes and redirections

Nicolas Delanoue

Université d'Angers - Polytech Angers



POLYTECH[°]
ANGERS



Definition

A process is the instance of a computer program that is being executed by one or many threads.

Remarks

- A process is not a program or a command
- A single command can give rise to several processes.

Different processes type

Interactive process Launched by a user via a command or the graphical interface, eg `ls` or `firefox`.

Automated process Launched automatically or programmed from a terminal, e.g. : `cron`, `at`

Daemon or service Process that runs continuously launched mainly at startup, eg web server, `ssh` server, `dots`

Lightweight process Process which runs under the authority of a main process via multitasking : the `firefox` tabs,

Kernel process Kernel tasks which ensure the proper functioning of the system : scheduler, hardware access, ...

Characteristics of a process

- PID : an identification number
- PPID : the PID of the parent process
- user who initiated the process
- its priority
- its state :
 - R (running),
 - S (sleeping),
 - T (stopped),
- space in memory, terminal, launch time . . .

A few comments :

- Each process has a father . . .
- all, except `systemd` (the first process)
- To see the hierarchy of processes, you can use the command `ps tree`
- Note for the teacher : demonstration of `ps aux`.

The command pstree

```
Terminal - nico@nico-OptiPlex-790: ~
Fichier Éditer Affichage Terminal Onglets Aide
nico@nico-OptiPlex-790:~$ pstree
systemd├─ModemManager├─{gdbus}
│                   └─{gmain}
├─NetworkManager├─dhclient
│                ├──dnsmasq
│                ├──{gdbus}
│                └─{gmain}
├─accounts-daemon├─{gdbus}
│                └─{gmain}
├─acpid
├─agetty
├─avahi-daemon├─avahi-daemon
├─colord├─{gdbus}
│       └─{gmain}
├─cron
├─cups-browsed├─{gdbus}
│             └─{gmain}
├─cupsd├─dbus
├─dbus-daemon
├─irqbalance
├─lightdm├─Xorg├─{InputThread}
│         │   ├──2*[{disk_cache:0}]
│         │   └─2*[{radeon_cs:0}]
│         └─lightdm├─upstart├─Thunar├─{gdbus}
```

ps is an system-monitor process-viewer.

```
Terminal - nico@nico-OptiPlex-790: ~
Fichier Éditer Affichage Terminal Onglets Aide
nico@nico-OptiPlex-790:~$ ps
  PID TTY          TIME CMD
 25343 pts/7    00:00:00 bash
 25353 pts/7    00:00:00 ps
nico@nico-OptiPlex-790:~$ ps u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
nico    16674  0.0  0.0  22780  5496 pts/2    Ss+  nov.15   0:00 bash
nico    16775  0.0  0.0  22932  5768 pts/6    Ss+  nov.15   0:00 bash
nico    25343  0.6  0.0  22648  5216 pts/7    Ss   15:58   0:00 bash
nico    25354  0.0  0.0  37472  3360 pts/7    R+   15:58   0:00 ps u
nico@nico-OptiPlex-790:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0 185796  5964 ?        Ss   oct.26   0:07 /lib/systemd/s
root         2  0.0  0.0     0     0 ?        S    oct.26   0:00 [kthreadd]
root         4  0.0  0.0     0     0 ?        I<   oct.26   0:00 [kworker/0:0H]
root         6  0.0  0.0     0     0 ?        I<   oct.26   0:00 [mm_percpu_wq]
root         7  0.0  0.0     0     0 ?        S    oct.26   0:01 [ksoftirqd/0]
root         8  0.0  0.0     0     0 ?        I    oct.26  19:22 [rcu_sched]
root         9  0.0  0.0     0     0 ?        I    oct.26   0:00 [rcu_bh]
root        10  0.0  0.0     0     0 ?        S    oct.26   0:00 [migration/0]
root        11  0.0  0.0     0     0 ?        S    oct.26   0:01 [watchdog/0]
root        12  0.0  0.0     0     0 ?        S    oct.26   0:00 [cpuhp/0]
root        13  0.0  0.0     0     0 ?        S    oct.26   0:00 [cpuhp/1]
root        14  0.0  0.0     0     0 ?        S    oct.26   0:02 [watchdog/1]
```

To stop a process (i.e. to kill it), you can choose one the following :

- clicking on the cross at the top right on the gui,

To stop a process (i.e. to kill it), you can choose one the following :

- clicking on the cross at the top right on the gui,
- asking it kindly :

```
kill <PID>
```

To stop a process (i.e. to kill it), you can choose one the following :

- clicking on the cross at the top right on the gui,
- asking it kindly :
`kill <PID>`
- without asking its opinion :
`kill -9 <PID>`

To stop a process (i.e. to kill it), you can choose one the following :

- clicking on the cross at the top right on the gui,
- asking it kindly :
`kill <PID>`
- without asking its opinion :
`kill -9 <PID>`
- by indicating its name rather than its PID :
`pkill <nomcommande>`

Remark

To make this working, you have to be the owner of the process (or root).

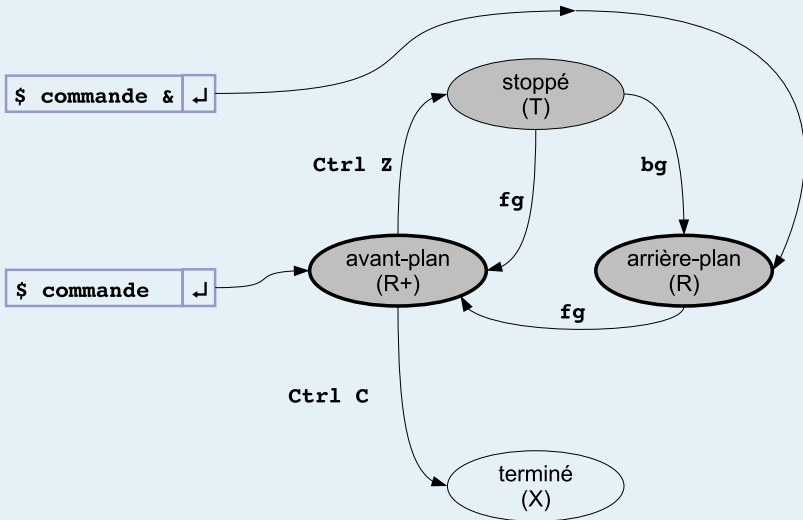
Definition

When we launch a process in a console, the console remains blocked until the process ends. The process is in *foreground*.

Remark

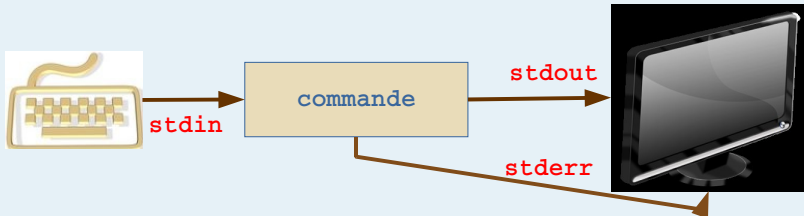
- You can ask the process to stop by typing `Ctrl C`.
- Start the background process to continue using the console while waiting :
`$<commande> &`
- Pause a process started in the foreground by typing `Ctrl Z`.
- To relaunch it
 - in background : `bg`,
 - in foreground : `fg`.

Activities and pauses in a process



By default, one command opens 3 streams :

- `stdin`, for standard input,
- `stdout`, for standard output,
- `stderr`, for standard error.



Definition

Redirect means replacing one of these streams by a file or by entering/exiting another command.

Redirection synthax

- < redirect the standard input,
- > redirect the standard output,
- >> redirect the standard output (adding),
- 2> redirect the error output,
- \&> redirect both the standard output and the error output.

Example :

<code>ls > liste</code>	create/overwrite a file liste and redirect the output of ls to the file liste
<code>date >> fichier.txt</code>	add at the end of the file fichier.txt the output of the command date date
<code>wc -l < f2.txt</code>	select as input to the command wc the file f2.txt

Definition

A pipe allows to connect to commands.

Example

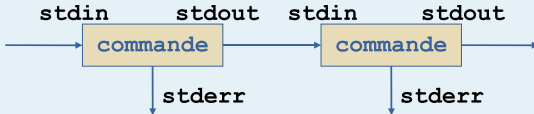
The command sequence

```
ls > temp.txt ; wc -l < temp ; rm temp
```

is equivalent to

```
ls | wc -l
```

Illustration



Filters

- `cat`
 - displays the content of files passed as parameters (for default, `stdin`)
 - options `-b`, `-n`, `-v`.
- `more`
 - displays page by page the files passed in parameters (default, `stdin`)
 - `h` to obtain help
- `less`
 - as `more` but backtracking possible
 - `q` to quit

Example

```
ps aux | more
```

The filter grep

- The `grep` software searches, in the file passed in parameter, the lines matching a given regular expression.
- syntaxe : `grep expr_reg [file]`

Example

- `grep "toto"essai`
search in test all lines that contain the word toto.
- `ls -l | grep network`
look in the result of `ls` all line that contains network.