Process management in UNIX/Linux system

Krzysztof Kluza

22.11.2021

Managing processes

1. Using `ps` list the processes working in the system:
   (a) Try to use various options:
   – using standard syntax, e.g. `-l, -e, -u` or complex, e.g. `-ejH`) or
   – arguments (using BSD syntax, e.g. `ax, axu, eo, f, l`).
   (b) How to list all the processes in the system?
   (c) Find the process that uses the most memory.
   (d) Which process consumes the most CPU?
   (e) Which are your processes in the system?
2. Using `kill`, one can send a signal to a process.
   How to list all possible signals?
3. Start a process in one terminal and try to terminate it using `kill` from the other terminal?
4. Tasks, are a mechanism provided by the shell to facilitate the handling of processes run in a given session (shell instance). You can run several processes in a given session, e.g.:

```
> lynx www.agh.edu.pl
[click Ctrl+Z]
> lynx www.uj.edu.pl
[click Ctrl+Z]
> vi &
> jobs
> jobs -l
> ps
> fg %2
[click Ctrl+Z]
> bg %1
> jobs -l
> kill %3
> jobs
```

5. Use `/tmp/datecont` program that generates the current time and date every second on the standard output.

   (a) Run the above mentioned program in the background by redirecting the standard output to a file.

   (b) Check if new data appear in the file.

   (c) Restore the task to the foreground (`fg`).

   (d) Stop it and check if the data is still being added to the file?

   (e) Enable the task in the background (`bg`), and check if the data is being added to the file?

   (f) Log in to the second terminal, try the SIGSTOP and SIGCONT signals on the above mentioned task (find the PID of this task).

   (g) Repeat the exercise without redirecting the output to a file.

6. How to run a process with a lower priority (higher `nice` value)?

7. (optional) System X Window If you are able to log in with X Window forwarding (using option `-x`) or you use your own Linux system, try running a window application `xeyes`, and try to kill it using `xkill`.

8. Using `screen`, run in 3 windows: `vi`, `man screen`, `watch date`.

   (a) Switching between the windows: write something in vi, read the man, see how the watch results change. Watch out for keyboard shortcuts (C-a is a prefix for screen commands)!

   (b) Detach the screen, so that the windows (session) do not close.

   (c) Log out of the server.

   (d) Log in again.

   (e) Check if you have active screen session, connect to an active session.

   HW Configure the screen to always display information about open windows in the last line.

   HW Screen is currently less frequently used, so try at home using: `tmux`.

9. Some additional possibilities of executing commands:

   - to execute commands in batch: `command1; command2`
   - background execution of command1: `command1 &`
   - concurrent execution of two command1 and command2: `command1 & command2`

   Test the above-mentioned executing methods with various commands and observe the task (job) status.