ANNEX A · PRACTICAL 5.5 · PROCESSES

Name: _____ **Date:** _____ **Group:** _____

Work in `deb-lab` (tasks 1–9) and `win11` (task 10). Hand in: this sheet, and on Classroom the report `practical-5.5-surname.md` with commands and outputs of tasks 7–10. Marking: tasks 1–6: 4 · task 7: 1.5 · task 8: 1.5 · task 9: 1 · task 10: 1 · clear report: 1.

PART A · LOOK

Task 1 · Count. Install the tools, then count processes.

```
sudo apt install htop psmisc
ps
ps aux | wc -l
ps -u $USER | wc -l
```

`ps` alone shows _____ processes. In the whole system: _____ Yours: _____ Who owns PID 1? _____

Task 2 · Read. Run `ps aux | head -5` and `pstree -p | head -15`.

Name of PID 1: _____ Column with the state: _____ Which process is the parent of your `bash`? _____

DONE WHEN you can follow the tree with your finger from `systemd` to your shell.

Task 3 · Live. Run `top`. Press `P`, then `M`, then `q`. Then run `htop` and leave with `F10`.

`P` sorts by _____ `M` sorts by _____ Process using most memory: _____

PART B · JOBS AND SIGNALS

Task 4 · Background. Start two processes that only wait.

```
sleep 300 &
sleep 400 &
jobs -l
ps -o pid,ppid,stat,comm
```

Job 1 has PID _____ Its PPID is _____ which is the PID of _____ State letter: _____

Task 5 · Pause, continue, end. After each line run `jobs -l` and write the word it shows for the job.

Command	`jobs -l` says	Signal sent (name)
`kill -STOP %1`		
`kill -CONT %1`		
`kill %1`		
`fg %2` then press `Ctrl+Z`		
`bg %2`		

DONE WHEN job 1 is gone and job 2 is running in the background. Finish with `kill %2`.

Task 6 · The stubborn process. Create this script in `~/lab`, make it executable and run it.

```
#!/bin/bash
trap 'echo "TERM received: saving and leaving"; exit 0' TERM
trap '' INT
echo "stubborn started with PID $$"
while true; do sleep 1; done
```

```
chmod u+x stubborn.sh
./stubborn.sh
```

1 Press `Ctrl+C` three times. What happens? _____

2 Open a **second** terminal and run `kill` followed by the PID. What does the first terminal print?

_____

3 Start it again. From the second terminal: `kill -9` and the PID. What does it print now? _____

4 Which of the two would you want for a program that is saving your work? Why?

_____

DONE WHEN you have seen the script ignore one signal, obey another and die without a word under the third.

PART C · THE RUNAWAY PROCESS (TRAMUNTANA APPS)

Marta's laptop has the fan at full speed. Something is eating the CPU.

Task 7 · Find it. Start the runaway, then find it with `top` (press `P`).

```
yes > /dev/null &
top
```

PID: _____ `%CPU`: _____ State (`S` column): _____ `NI`: _____ Owner: _____

Task 8 · Calm it down. Make it the last in line. Then try to give it back its priority.

```
renice -n 19 -p PID
renice -n 0 -p PID
```

First command says: _____

Second command says: _____

Is `%CPU` lower now? _____ Why (not)? _____

Task 9 · Stop it properly. Polite first. Then check that it is gone.

```
kill PID
jobs
pgrep -a yes
```

`jobs` says: _____ `pgrep` prints: _____

Start a second one with `nice -n 10 yes > /dev/null &` and end it by name. Command: _____

DONE WHEN `pgrep -a yes` prints nothing.

Task 10 · The same on Windows. In `win11`, in PowerShell:

```
Start-Process notepad
Get-Process notepad
Stop-Process -Name notepad
Get-Process | Sort-Object CPU -Descending | Select-Object -First 5
```

`Id` of Notepad: _____ Process with most CPU time: _____ Keys that open Task Manager: _____

DONE WHEN Notepad closes from the command line and you have named the top process.

EXIT TICKET (NO VM, ALONE)

1. What is a process?

- ☐ **a)** A program stored on the disk
- ☐ **b)** A program that is running, with a PID
- ☐ **c)** A file with execute permission
- ☐ **d)** A window on the screen

2. Which signal does `kill 4636` send?

- ☐ **a)** SIGKILL (9)
- ☐ **b)** SIGINT (2)
- ☐ **c)** SIGSTOP (19)
- ☐ **d)** SIGTERM (15)

3. Why is `kill -9` the last thing to try?

- ☐ **a)** The process cannot save or clean up
- ☐ **b)** It needs `sudo` every time
- ☐ **c)** It also ends the parent process
- ☐ **d)** It only works on background jobs

4. What does the `&` in `sleep 300 &` do?

- ☐ **a)** Runs it twice
- ☐ **b)** Runs it as root
- ☐ **c)** Runs it in the background
- ☐ **d)** Runs it with low priority

5. A process has nice value 19. What does that mean?

- ☐ **a)** It has the highest priority
- ☐ **b)** It is last in line when the CPU is busy
- ☐ **c)** It may use only 19 % of the CPU
- ☐ **d)** It ends after 19 seconds