

Processes and Job Control

1. Start the `nano` from the terminal and type a few lines. Suspend the `nano` process using `Ctrl+Z`.
2. Use `ps` (with some options) to get the PID and PPID of the `nano` process. Can you relate the `nano` process to one of the other processes?
3. Use `jobs` to find your `nano` process. What is its running status?
4. Kill the `nano` process using its PID. What is its running status now in the job list?
5. Start a new `nano` process in the background.
6. How can we activate this newly created process?
7. Execute the command: `$ yes > /dev/null &`. What does it do?
8. Inspect the running `yes` processes interactively using `top`. How much resources (CPU, memory) does it consume? Don't forget to stop the process in the end.
9. Execute `$ yes | grep 'y' > /dev/null &`. Inspect with `top`. What could go wrong in a shared multi-user environment? If we terminate the `yes` process, what happens to the `grep` process?
10. Try to set up an environment where you can simultaneously 1) run a script 2) edit a script 3) monitor its progress. (Tip: use one of yesterdays assignments).
11. Some commands can take a long time. Time the command: `$ du -h /`. Why do the times not add up?
12. Time the command: `$ du -h / > /dev/null`. Why is there still output? Explain the time difference to the previous run.