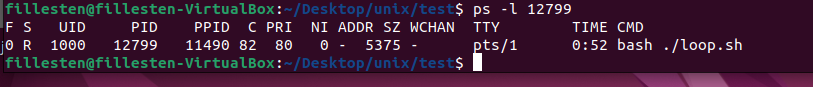
Admin av UNIX LABORATION 2

## 4.1

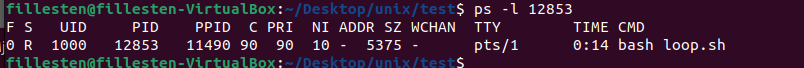
En bild som visar text

Automatiskt genererad beskrivningPID of the looping process is 11847

It has nice value of 0 (default)



After I increased it.



En bild som visar text

Automatiskt genererad beskrivningHere I try to increase the nice value:  
without using sudo I cant lower the nice value (increase priority) of a process.   
and with sudo I will be very likely to change it.

I may have tried different commands and have to restart it with another PID…

En bild som visar text

Automatiskt genererad beskrivningEn bild som visar text

Automatiskt genererad beskrivningKilled it using kill command

En bild som visar text

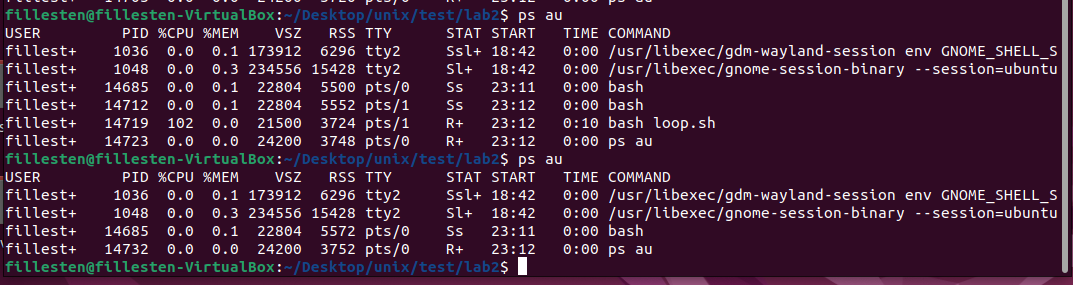
Automatiskt genererad beskrivningKilled using pkill command

* Nohup:

En bild som visar Webbplats

Automatiskt genererad beskrivningEn bild som visar text

Automatiskt genererad beskrivning

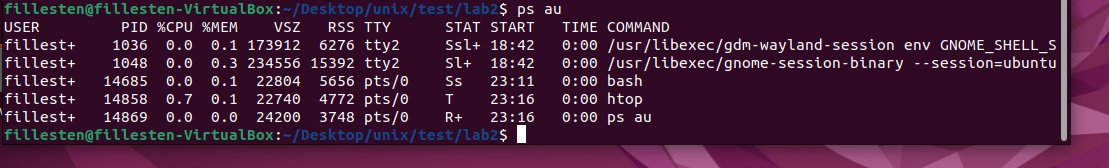


The script loop.sh is not running after using the nohup command.

* Using the & character

En bild som visar text

Automatiskt genererad beskrivning



It does not recognize that the process is there, but using top I can locate the process (processes as I accidentally started a few)

En bild som visar kalender

Automatiskt genererad beskrivning

* Jobs:

En bild som visar text

Automatiskt genererad beskrivning

When stopped:

En bild som visar text

Automatiskt genererad beskrivning

Then I start it in the background:

En bild som visar text

Automatiskt genererad beskrivning

En bild som visar text

Automatiskt genererad beskrivningWeirdly I could not start it as a background process but that is how you are supposed to do it. When I tried the bg %1 command I lost control of the terminal, it continuously printed out like “normal” and I could not use CTRLZ or CTRLC. However I could still kill it from another terminal, which resulted in this… But here it doesn’t look like it was a background process.

**Answers**

* *Priority value* is what the process priority behind the scenes actually are. Nice value is only user-space that allow users to control to a degree the priority of processes.
* -20 is the highest priority, +19 is the lowest priority.
* There are multiple ways to To start a process in the background
  + you can append the “&” character; ./loop.sh &
  + use the nohup command; nohup bash loop.sh &
  + bg %<jobnumber>

to return a process to the foreground the command “fg”, the fg command can be done on running processes.

* Nohup is often used when running long time running or for background processes that shouldn’t get interrupted if the user logs out or if the terminal closes. Example of these jobs are: running a server of running a large script job that takes a long time.

I had to create a new ubuntu since I cant start the first one. /dev/sda3 is full somehow and I tried to fix it but have since given up.

## 4.2

1. Create the scriptEn bild som visar text, skärmbild, Teckensnitt

   Automatiskt genererad beskrivning

Make it executeable

En bild som visar text, skärmbild, Teckensnitt

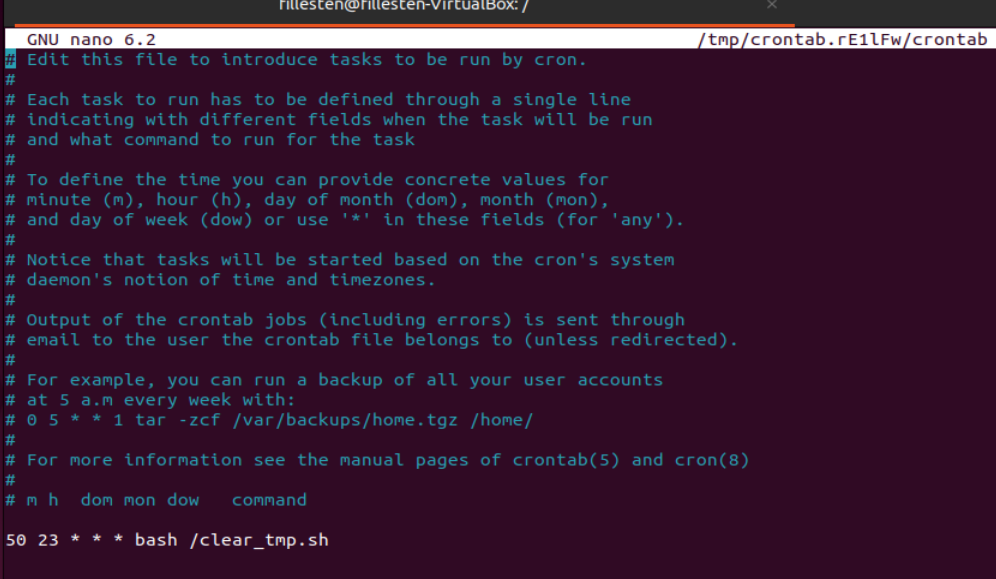
Automatiskt genererad beskrivning

Add some files to remove with the script.

En bild som visar text, skärmbild, meny

Automatiskt genererad beskrivning

1. Configure crontab to run the script every evening at 23.50



1. Running the script at 21.30 using at

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivning

* The difference between crontab and at is that crontab can be scheduled to execute more than once, like automatically searching for updates. The at command only lets users run a command once.

## 4.3

1. Produces this:

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivning

* 1 runnable process which is either running or waiting for cpu time.
* 0 processes in sleep mode
* Memory: (kilobytes)
  + Swpd: 0 virtual memory used
  + Free: 1718412 amount of free memory available for new processes.
  + Buff: 48108 memory used for buffering
  + Cache: 957032 memory used for caching
* Swap:
  + si: Amount of memory swapped in from the disk.
  + so: Amount of memory swapped out to the disk
* io:
  + bi: 153 Blocks received from block devices
  + bo: 18 Blocks sent to block devices.
* System:
  + in: 93 interrupts per second, including clock interrupts.
  + cs: 150, Number of context switches per second.
* Cpu:
  + us: 1, Percentage of CPU time spent running user processs
  + sy: 1, Percentage of CPU time spent running kernel processes.
  + id: 98, Percentage of CPU time spent idle.
  + wa: 0, Percentage of CPU time spent waiting for I/O.
  + st: 0, time stolen from a virtual machine

1. top updates every few seconds.

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivning

I get the current time when top retrieved the data. Amount of users. Load average with 3 numbers which are: 1, 5, 15 min ago respectively.   
**Tasks row**: It lists the total amount of tasks (210 for me), where 1 is running and 209 are sleeping. 0 tasks are stopped and 0 are in a zombie state.

**Cpu row**: percent of time spent on different tasks. Us:  is the percent of time spent running user processes, sy: is the percent of time spent running kernelspace processes, ni: spent on tasks which have had their nice values manually altered, id: is the percent of time idle, wa: is the percent of wait time, hi: is the percent of time managing hardware interrupts, si: is the percent of time managing software interrupts, st: is the percent of virtual CPU time waiting for access to physical CPU.

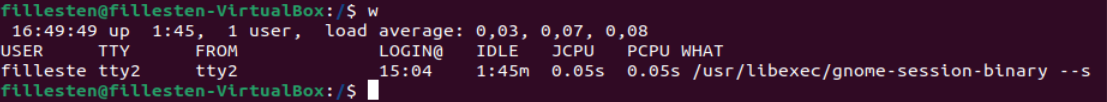
**Mib mem (physical)**:   
total: shows total installed memory, free: shows available memory, used: shows used memory,   
buff/cache: shows the information to be cached for future reads

**Mib swap (virtual)**:  
same as above but for virtual memory.

**Columns**:  
PID: process id  
USER: owner of task  
PR: priority of the task  
NI: nice value of the task  
VIRT: total virtual memory used by task  
RES: total physical memory used by task  
SHR: total shared memory used by task  
S: shows the process state, like sleeping zombie etc   
%CPU: CPU usage  
%MEM: memory usage  
TIME+: CPU time, the units are min:sec.fraction   
COMMAND: name / description of the task

1. 813,5 MiB used and 0,0 Mib swap usedEn bild som visar text, skärmbild, Teckensnitt

   Automatiskt genererad beskrivning
2. Last 10 logged in users:  
   En bild som visar text, skärmbild, Teckensnitt

   Automatiskt genererad beskrivning
3. Current logged in users:  
   
4. Using the uptime command I get this:  
   En bild som visar skärmbild, text, Teckensnitt, Grafik

   Automatiskt genererad beskrivning

The output is, current time, how long the system has been active, # of users logged in, and load average of 1, 5, 15 minutes.

* The difference between vmstat and top are the outputs. vmstat provides system-level statistics on memory, CPU, I/O, and paging, while top is a real-time process viewer showing individual process details, including CPU and memory usage. vmstat is for system-wide resource monitoring, and top is for monitoring and managing running processes.
* The different load average values are because they represents the average regarding different times. The are 1, 5, 15 minutes.
* Last 10 users:  
  En bild som visar text, skärmbild, Teckensnitt

  Automatiskt genererad beskrivning

## 4.4

1. This is a snapshot of the last info from dmesg command. It contains the messages produced by device drivers. All problems during start up will be displayed here.

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivning

1. There are multiple logs here:

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivning  
auth.log contains messages regarding authentication and authentication events.

1. Sending a message to syslog.  
   En bild som visar text, skärmbild, Teckensnitt

   Automatiskt genererad beskrivning

* syslog is a system-wide logging system for storing messages from various sources. dmesg displays kernel-related messages and is used for diagnosing hardware and kernel issues. syslog is for general system and application logs, while dmesg is focused on the kernel's messages and boot process.
* Proof it received it:

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivning