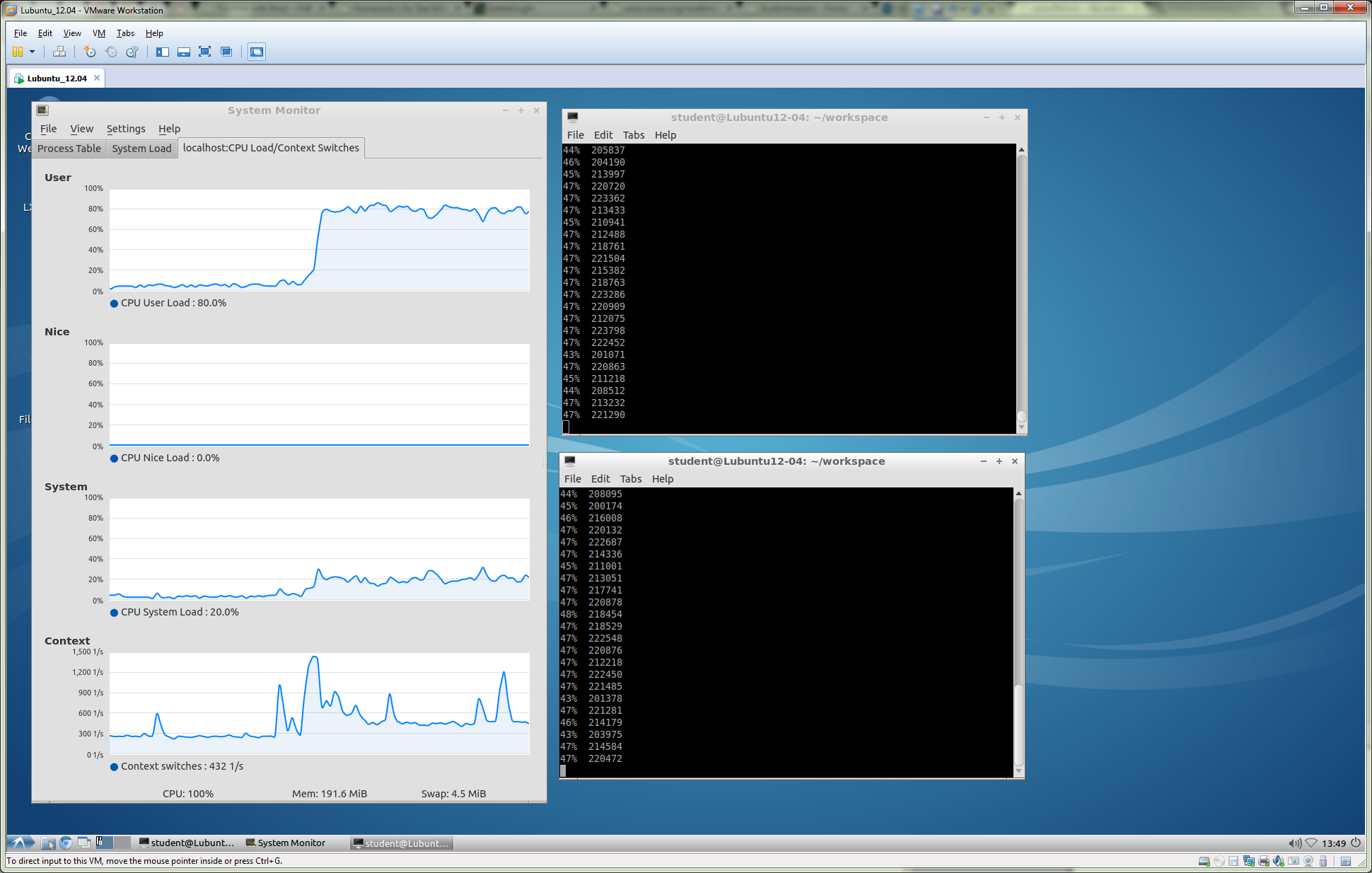
# Opdracht ABC

Joery van den Hoff

## Opdracht A

Hier een overzicht van mijn bevindingen per type programma. Op de screenshot is mijn opzet te zien.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **User load** | **Nice load** | **System load** | **Context switches** |
| **User-busy** | Continue erg hoog | Continue 0 | Vrij laag | Iets hoger dan normaal |
| **System-busy** | Iets hoger dan normaal | Continue 0 | Continue erg hoog | Iets hoger dan normaal |
| **Sleeping** | Erg laag | Continue 0 | Erg laag | Erg hoog(6x) |
| **Scheduling** | Erg laag | Continue 0 | Continue erg hoog | Extreem hoog(1200x) |

**Low priority**

Als ik het user-busy start met het volgende commando nice –n 19 [..] dan heeft het proces een lage prioriteit. (Het nummer gaat van 19 tot -20 voor lage tot hoge prioriteit respectievelijk.)

De nice load in KSysGuard gaat dan naar 80%.

## Interesting scenario’s

Als ik hetzelfde programma met een hoge en een lage prioriteit run dan wordt de CPU alsnog eerlijk verdeeld, dat vind ik raar. Ook met 3 hoge prioriteit en 1 lage prioriteit, krijgt de lage prioriteit een kwart van de processor tijd. Misschien komt het omdat de host machine (Windows) prioriteiten niet ondersteund voor de VM.

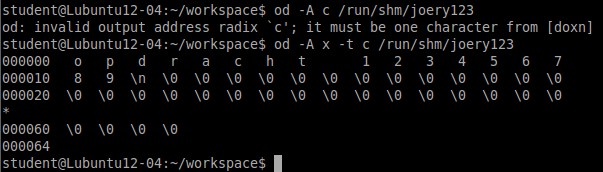
## Numbers

Het eerste nummer is de CPU percentage dat het huidige proces gebruikt heeft (sinds de laatste seconde). Het tweede nummer is het aantal cycles dat doorlopen zijn sinds de laatste seconde.

Counter wordt elke keer opgehoogd als er een cycle is geweest en gereset als er een seconde voorbij is.

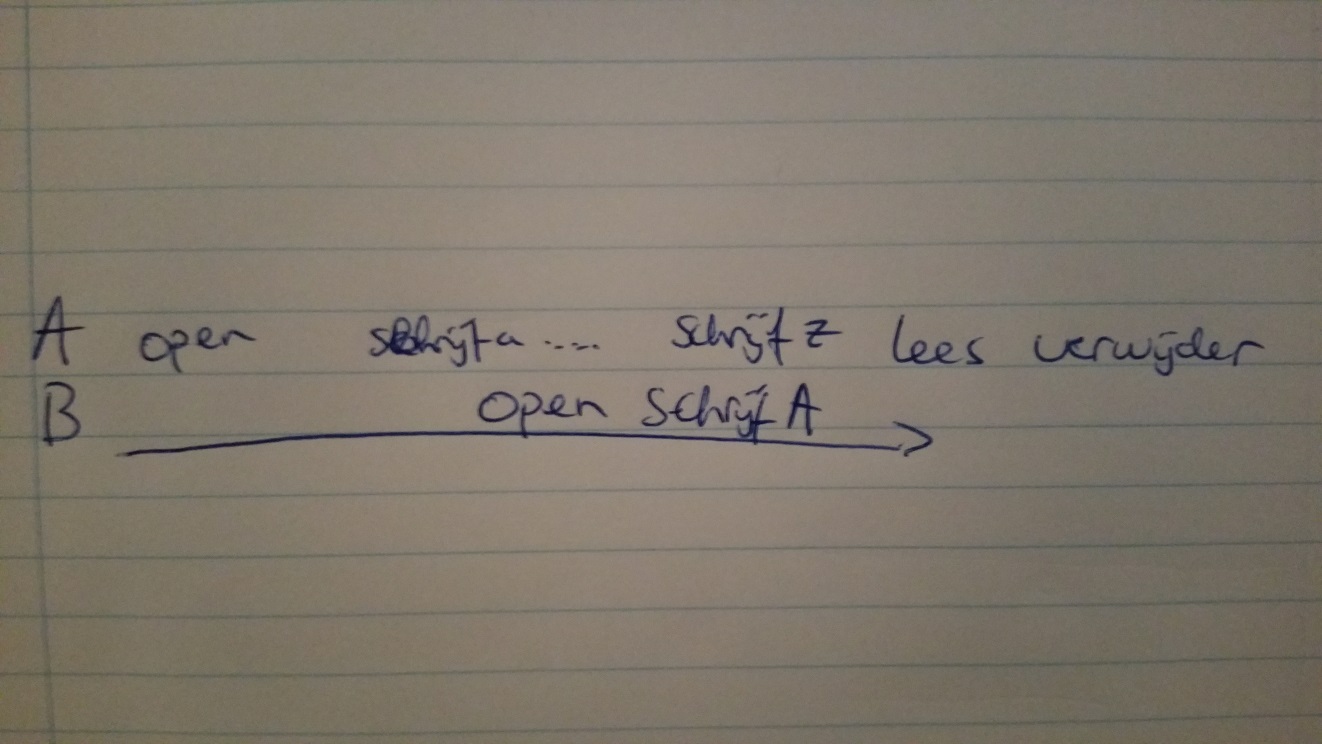
## Opdracht B

* Met de lseek() functie kan de grote van een file bepaald worden door SEEK\_END als whence mee te geven.
* Mijn shared memory genaamd ‘joery123’ is te vinden onder /run/shm/joery123.
* Met het commando ‘od –A x –t c /run/shm/joery123’ print hij netjes de tekst uit die ik erin had geschreven (‘opdracht 123456789’).



## Programma’s

Ik heb twee programma’s gemaakt zoals beschreven in de manual. Zie programa.cpp en programb.cpp. De vragen:

* 
* Proces B kan nog steeds lezen en schrijven omdat shm\_unlink het niet verwijderd tot alle referenties zijn verwijderd.
* Twee processen kunnen met elkaar communiceren door afspraken te maken, wanneer er gelezen/geschreven dient te worden. Bij deze programma’s was de afspraak dat zodra de eerste letter een a is, de eerste letter aangepast kan worden.

## Opdracht C

Server parameters

* -p 1234 betekent dat de server op poort 1234 gaat luisteren
* -d betekent dat er een seconde gewacht wordt voor dat het bericht wordt terug gestuurd
* -v betekent dat er veel meer informatie wordt uitgeprint over wat er aan de hand is (verbose)
* -g staat voor debug en houdt in dat alle opties en de compiler versie worden uitgeprint

Client parameters

* -i 192.168.0.103 is het ip waarmee verbinding gemaakt wordt
* -p 1234 is de poort waarmee verbinding gemaakt wordt
* -d, -v, -g zie hierboven
* aBcDeFgHiJkLmNoP 1234 XYz is de data die verstuurd zal worden.

Ik heb de TCPEcho applicatie afgemaakt. Hieronder is hij in werking te zien.

