

## VE482 — Introduction to Operating Systems

### Homework 2

Manuel — UM-JI (Fall 2019)

Non-programming exercises:

- Write in a neat and legible handwriting
- Clearly explain the reasoning process
- Write in a complete style (subject, verb and object)

Programming exercises:

- Write a README file for each program
- Upload an archive with all the programs onto Canvas

#### Ex. 1 — Multiprogramming

A few years ago when computers featured less RAM it is was common to increase it in order to enhance CPU performance. In order to better understand the link between the two we now create a simple model for multiprogramming. We assume all the processes to be similar and spending the same fraction  $p$  of their time waiting for Input/Output (I/O) to complete.

1. What is the probability for  $n$  processes to be waiting at the same time, then express the CPU utilisation as a function of  $n$ ?
2. Sketch the curve representing the CPU utilisation as a function of the number of processes for the following values of  $p$ : 25%, 60% and 90%.
3. A certain old computer has 256 MB of RAM, once loaded a light operating system uses 96 MB of RAM. Several programs are launched each of them using 48 MB.
  - a) How many processes can be store simultaneously in memory?
  - b) Assuming an average of 90% I/O waiting time what is the CPU utilisation?
  - c) What is the effect of adding 256 MB, 512 MB and 1024 MB of RAM. Argue on which amount would be the most beneficial and would be worth the investment.

#### Ex. 2 — Keymap in Minix 3

Map the Shift+F7 key to displaying how many processes are currently running.

This can be achieved following the steps:

1. Find all the files that need to be modified;
2. Locate the part of the code that should be updated;
3. Check how to process table is stored;
4. Count the number of running processes;
5. Map the newly written function to the Shift-F7 key;
6. Recompile the kernel and test.

Refer to Minix user's guide for some basic help.