

Lab and Homework #4

Introduction to Operating Systems CS-UY 3224 | CS-UY 3224G

Mirna Džamonja, email md5961@nyu.edu

Due date for the Homework problems : October 2nd, 2023 by 5 PM, Paris time

Please hand in through the *Assignments* option on *Brightspace*.

Question 1: *A bit of commands in Unix/Linux.*

Please work on this question from the Terminal.

- Check the paging situation on your computer by doing `vmstat`. Familiarise yourself with this command by reading its `man` page.
- Run the `top` command. What does it do (run the `man`)? What information does it give you about memory?

Nothing to hand in for this exercise.

Question 2: *Dynamic allocation of memory in C.*

In this exercise you are asked to write a C program that determines the prime numbers up to a number given by the user. The program will have to dynamically allocate the memory needed for the task. You should use the function `malloc` of C.

To do at home and hand in: Your C program.

Question 3: *Memory allocation algorithms.*

Here we are practising the memory algorithms that we learned. The calculations might be a bit time-consuming, but they are simple. The assumptions that we make are that:

There are four jobs, A, B, C, D such that:

- The jobs have already been segmented, so every job has to fit in a single continuous chunk of memory.
- If needed, MMU has the right to stop and relocate jobs in order to compact memory.

Suppose that you have the following situation, in a memory allocated between addresses 0 and 3400, as follows:

A takes the addresses 0-500, B takes 900-1600, C takes 1900-2300, D takes 3100-3400.

The following actions happen, in this order:

- E starts and requests 300 units
- A requests 300 more units
- D exits

- F starts and requests 700 units
- C exits
- G starts and requests 900 units.

Give the contents of the memory under the following allocating algorithms:

- first-fit
- best fit
- worst-fit

I suggest to make a graphic representation first, in the style of OSTEP16-17 or other, as then the calculations become easier to visualize.

To do at home and hand in: All the work for this problem. What is the best algorithm for this example?

Question 4: *Git*.

A bit more of `git`. Do the exercise on the git tutorial site.

https://kbroman.org/github_tutorial/pages/fork.html

To do at home and hand in: Give me a link/access to what you have done.