

## Laboratory 2

### Part I

Print the string “Hello World” on screen. Each character must be printed by a different process. The process that prints the  $i^{\text{th}}$  letter must have been spawned by the process that printed the  $(i-1)^{\text{th}}$  letter. Each process must first print its designated character, as well as its own process ID, second, sleep for a random number of seconds (from 1 to 4 seconds), and then, do anything else it must do to achieve the given task.

What is the minimum lines of C code with which you can achieve the above?

Remember these processes have to be run in the guest Minix3 system. To compile in the guest system, we use a C compiler named *clang* (instead of *gcc*). Also, standard *make* is invoked as *gmake* in the Minix3 environment.

Submit: a single zip file (format: <roll-number-1>\_<roll-number-2>\_lab2\_part1.zip) with all required source files and a Makefile. The evaluator will simply unzip the submission, and run “gmake hello”. If the desired output is not seen, you will not be awarded any marks.

### Part II

Modify the Minix3 source code such that:

- A message “Minix: PID <pid> created” is printed, whenever a process is created. (Let us follow the convention throughout this course that anything printed by the Operating System code will be prepended by the string “Minix: “.)
- A message “Minix: PID <pid> exited” is printed, whenever a process ends.

Comment on the order in which processes are created and processes exit and justify it is as expected.

Submit: a single zip file (format: <roll-number-1>\_<roll-number-2>\_lab2\_part2.zip) with all modified source files and a shell script. The shell script must copy the modified source files to the correct directories, and build the system. The evaluator will simply run the shell script, reboot the system, and check for the desired behavior.

Hint: look at *minix/servers/pm* .

Important: Remember to create a branch in the git repository for this assignment.