Assignment 1

NAME____

Write your name above.

Grading Rubric

There are five questions in this assignment, each worth the percentage indicated. Questions are assessed on their completeness and correctness and graded in accordance with the ***General Requirements for Assignments*** document posted on the course website.

Instructions for Submitting the Homework

1. **Due Date**: ***This assignment is due in class on February 24.***
2. Your homework must comply with the ***General Requirements for Assignments*** document posted on the course website for full credit. Submissions that do not follow these instructions will not be graded and will be given a **zero grade**.
3. ***Print this assignment single-sided and fill in your answers in the space provided. If you do not print it single-sided, it will not be accepted.***
4. Bring the completed assignment to class. It will not be accepted after the end of the class on February 24.

About the Questions

All of these questions are related to material from Chapters 1, 2, and 3. One purpose of this assignment is to get you to do some conde-reading. Another is to get you to see how practical questions about operating systems are handled in a real operating system, namely Linux. Because Linux is a free and open source operating system, you can rummage through its code, and you can even customize it for yourself if you really know what you are doing.

I have cloned the most recent stable release of the Ubuntu kernel source code in the directory `/data/biocs/b/ubuntu/bionic/` on the Computer Science Department's server, `eniac.cs.hunter.cuny.edu`, to which everyone has access. You can login to `eniac`, then into any `cslab` host and navigate to that directory and browse its contents.

Questions

1. (15%) Linux makes it easy to explore the current state of the operating system and all of its processes through a ***pseudo-filesystem*** named `/proc`. The `/proc` filesystem is an interface to kernel data structures. If you login to any of the `cslab` hosts in out network, you can view the contents of various files in the `/proc` directory. One file there that rarely changes, is `cpuinfo`, which contains information about the cpus of the given host. Look at this file on the following hosts: `cslab12`, `cslab13`, `cslab14`, and `cslab15` and answer the following questions in the boxes below the question.

   (a) For each host, how many processors does it have?

   (b) For each host, what is the processor model name? (They are all the same in any given machine.)

   (c) For each host, how many kilobytes of capacity are in each core's cache?

2. (10%) The `/proc` directory has a directory for every running user process, named by its process id (*pid*), as well as a directory for the process with pid 1, which is in Linux is called either `init` or `systemd`, depending on the version of Linux that you have. It is the ancestor of all user processes. Every one of these directories in `/proc` has a file named `limits` that contains the process's resource limits. Answer the questions below in the boxes provided for the process with pid 1.

   (a) What is the maximum number of processes that it can have?

   (b) How many **kilobytes** is its maximum stack size soft limit?

3. (15%) Find the file `include/linux/sched.h` in the directory `/data/biocs/b/ubuntu/bionic/`. In that file, there is a definition of a structure named `task_cputime`. The comments explain what the members of this structure are.

   (a) Write the definition of that structure in the box below.

   (b) What is the largest value that each of the structure's members can have? You might need to look at the file `types.h` in this same directory to answer this question. Write that value in the box below.

   (c) In the box below, explain briefly what kinds of time specifically are stored in each member, i.e., time spent doing what? You might need to search elsewhere for an answer.

4. (30%) The `sched.h` file also has the definition of a `struct` that Linux uses for representing a ***process control block***. Find the part of this `struct` definition that represents the children of a process.

   (a) In the box below, write the part of the `struct` that defines the process's children.

   (b) In the box below, write the type of data structure used to represent the children? Be specific. You might need to look at the file `types.h` again.

5. (30%) Find the file `./include/linux/kernel.h` in `/data/biocs/b/ubuntu/bionic/`. Somewhere in this file is a definition of a type that stores the states in which the operating system can be at any moment of time. Find that type and answer the following questions:

   (a) Write the name of this type in the box below.

   (b) List in the box below, in order, the names given to the states in which the operating can be.