Name: ***(as it would appear on official course roster)***		
Umail address:	**@umail.ucsb.edu**	**section** **9am, 10am, 11am, 12pm**
Optional: name you wish to be called if different from name above.		
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone"		

1

h16

CS32 F19

h16: Intro to Operating Systems

ready?	assigned	due	points
true	Tue 11/19 02:00PM	Tue 12/03 02:00PM	50

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE LISTED ABOVE AS THE DUE DATE,
OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments;
in place of that, we drop the lowest scores (if you have zeros, those are the lowest scores.)

Reading: Intro to Operating Systems, CS32 Reader (p. 83 - 116)

1. (10 pts) Fill in the information in the header. The following are required to get the 10 "participation" points.
   - Filling in your name and umail address.

   Also: For paper submission PLEASE submit on ONE SHEET OF PAPER, double-sided if at all possible. If you must submit on two printed sheets write name on BOTH sheets and no staples, paperclips, or folded corners.

2. Section (1.2) describes a "bottom-up" approach and a "top-down" approach to describing the purpose of an operating system. From the standpoint of these two points of view, give the main "purpose" of an operating system in a single **short phrase—8 words or less**. Your answers should capture the main points of these two perspectives on an OS.
   a. (3 pts) bottom-up:

   b. (3 pts) top-down:

3. The history of Unix as an OS (and then, the history of all Operating Systems, including those derived from Unix, such as Linux, Mac OS X, etc.) was heavily influenced by the choice of programming language in which Unix was first implemented.
   a. (2 pts) What programming language was chosen for the first implementation of Unix?

   b. (2 pts) Most other OS implementations, up to that point in time, had been written in what language?

   c. (2 pts) According to the authors, as a *direct result of these two facts* (your answers to (a) and (b)), Unix had a distinct advantage over other OSes. What was this advantage?

2

h16

CS32 F19

4. The "machine cycle" (also called the "instruction cycle") is a core concept in understanding computing. In the description of this cycle in our book, the three steps start with the letters "f", "d", and "e".
    a. (3 pts) What I'm about to ask of you may seem like busy work, and I apologize for that—but please trust me: it's important. We need to lay a solid foundation for future courses in which your knowledge of this "machine cycle" is going to be assumed. So, please: find the diagram of the "instruction cycle", and copy the drawing of it into the box at right.

       Now, please briefly describe what each of these steps means. Each is labelled with the first letter of the step.

    b. (3 pts) f:

    c. (3 pts) d:

    d. (3 pts) e:

5. In the Unix system, the "system call interface" provides entry points into ... something.
    a. (3 pts) In a single word or short phrase (no more than four words) name what the system call interface provides an entry point into.

    b. (3 pts) Now, in a 2-3 sentences, describe what the phrase means. What is the "purpose" of the thing you named in part "a"?

6. A distinction can be drawn between volatile storage and non-volatile storage in a computer.
    a. (3 pts) Give an example of volatile storage in a computer.

    b. (3 pts) Give an example of non-volatile storage in a computer.

7. (4 pts) People often talk about adding RAM to speed up their computers, but RAM doesn't actually make the CPU run any faster. How does adding RAM speed up computation without making the CPU any faster? (Consider the difference between RAM and Disk Memory)