## CSC 262 Homework #1 Due at start of class on Wednesday, February 7.

1. Notable Computer Systems. Operating systems have not arrived at their current state overnight: there have been many steps along the way. Some of these prior efforts are famous within the computing community, and are worthwhile background knowledge for you as a student of computer science. Listed below are a number of historical computer systems that have achieved noteworthiness in one way or another. Write a sentence or two describing what was distinctive or innovative about each one in its time. Then try to place the system into one of the "generations" discussed in class. If you cannot find specific information about the OS used on a particular computer, then make an educated guess based upon its age and intended function. (This assignment used to require more extensive research before the advent of Wikipedia. Although generally anything found on Wikipedia should be read with a critical eye, for the purposes of this question you may assume that the articles there are authoritative.)

- a. UNIVAC:
- b. Whirlwind:
- c. IBM 7090:
- d. GE-645:
- e. Data General Eclipse MV/8000:
- f. PDP-11:
- g. Cray-1:
- h. Connection Machine:
- i. Apple II:
- j. Commodore Amiga:

2. Efficiency Computations. Consider the following definitions:

 $Utilization = \frac{time\_used}{total\_time\_elapsed}$  $Throughput = \frac{jobs\_completed}{total\_time\_elapsed}$ 

*Latency* = *total* \_ *time* \_ *elapsed* 

Given the sets of jobs described below, calculate the overall CPU utilization, throughput, and latency, as well as the latency of each individual process, under the conditions described in each section.

- **Job 1**: Requires 8 CPU minutes total. Assume that I/O and memory latency for this job limit CPU usage to a maximum of 80%. (Hence, running by itself it will finish in 10 minutes.)
- Job 2: 3 CPU minutes, 60% max.
- **Job 3**: 2.4 CPU minutes, 30% max.
- **Job 4**: 1.6 CPU minutes, 10% max.

a. Assume serial batch processing. In other words, the jobs are processed one at a time, starting with Job 1 and ending with Job 4. Unused CPU goes to waste.

b. Assume pseudo-parallel execution. In other words, jobs are scheduled when there is sufficient CPU available to run them at full speed. If multiple jobs are eligible to be run, the lowest job number is scheduled. (For example, all four jobs are eligible at the start, but Job 1 would be scheduled. With the remaining 20% of the CPU, Jobs 2 and 3 cannot be run, but Job 4 can. So Job 4 would run in parallel with Job 1. When either of those two finishes, other jobs may begin.)

c. Assume full multiprogramming. The microprocessor is divided as equally as possible between all active jobs. Jobs will run more slowly if they cannot get their full allotment of CPU time. For example, if Job 1 is allotted (say) only 20% of the CPU, then it will take 40 minutes to finish instead of 10. If it runs for 15 minutes at 20%, and then the allotment is inceased to 40%, it will require 12.5 more minutes to finish. (Hint: under the multiprogramming system, figure out how much CPU time each process gets at the start, and therefore how long it will be until one of the processes finishes. Figure out how much computation is left on each of the remaining processes at that point, reapportion the CPU, and continue.)