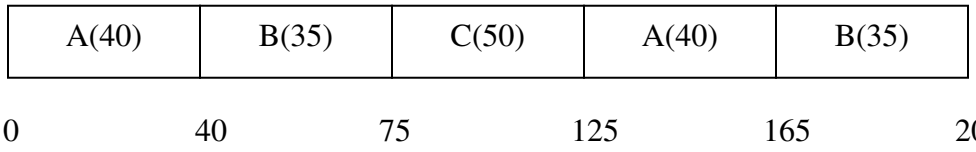
Cpr E 308, Spring 2006: Homework 3 Solution

1.

2. Yes. They are schedulable.



3. No. There is no deadlock.

Chapter 2

36. In simple cases it may be possible to determine whether I/O will be limiting by looking at source code. For instance a program that reads all its input files into buffers at the start will probably not be I/O bound, but a problem that reads and writes incrementally to a number of different files (such as a compiler) is likely to be I/O bound. If the operating system provides a facility such as the UNIX ps command that can tell you the amount of CPU time used by a program, you can compare this with total time to complete execution of the program. This is, of course, most meaningful on a system where you are the only user.

44. The fraction of the CPU used is 35/50 + 20/100 + 10/200 + x/250. To be schedulable, this must be less than 1. Thus x must be less than 12.5 msec.

Chapter 4

1. What problems would we face if there was no virtual memory in a PC Operating System?

If there was no virtual memory, we can only use physical address instead of virtual address, and keeping track of the exact physical addresses will lead to too much complexity. Second, in order to achieve multiprogramming without virtual memory, one solution is dividing the memory into fixed partitions, e.g. MFT, but the number of processes existing in the memory is fixed. Another solution is to use swapping technique to dynamically allocate memory, but the number of processes will be much smaller than using virtual memory. It is because swapping keeps the entirety of each process in the memory, running it for a while, and putting it back on the disk, while virtual memory strategy allows programs to run even when they are only partially in main memory.

2. Explain why, under normal circumstances, a Linux process cannot access the contents of another process's stack?

Because in Linux, different processes are mapped to different memory locations, e.g. each process has base and limit registers interpreting its memory space. Therefore, one process cannot access another process's stack.

6. Real memory uses physical addresses. These are the numbers that the memory chips react to on the bus. Virtual addresses are the logical addresses that refer to a process' address space. Thus a machine with a 16-bit word can generate virtual addresses up to 64K, regardless of whether the machine has more or less memory than 64 KB.

7. For a 4-KB page size the (page, offset) pairs are (4, 3616), (8, 0), and (14, 2656). For an 8-KB page size they are (2, 3616), (4, 0), (7, 2656).

8. (a) 8212 (b) 4100 (c) 24684

9. They built an MMU and inserted it between the 8086 and the bus. Thus all 8086 physical addresses went into the MMU as virtual addresses. The MMU then mapped them onto physical addresses, which went to the bus.

14. The number of pages depends on the total number of bits in a, b, and c together. And since a+b+c = total # bits - d, the number of pages depends only on d.

15. For a one-level page table, there are 232 /212 or 1M pages needed. Thus the page table must have 1M entries. For two-level paging, the main page table has 1K entries, each of which points to a second page table. Only two of these are used. Thus in total only three page table entries are needed, one in the top-level table and one in each of the lower-level tables.

16. The code and reference string is as follows

LOAD	6144,R0	1(I), 12(D)
PUSH	R0	2(I), 15(D)
CALL	5120	2(I), 15(D)
JEQ	5152	10(I)

The code (I) indicates an instruction reference, whereas (D) indicates a data reference.