CS 4760	**Operating Systems**	Test 3
Name:	Spring 2021	Max Pts: 64

Important: This is an open book test. You can use any books, notes, or paper. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem. The test is 75 minutes long.

1. [6 pt] How does paging facilitate copy-on-write technique?

2. [6 pt] What is the difference between a linear address and a physical address in the x86 architecture?

3. [6 pt] Consider a fixed partitioning scheme with equal size partitions of 2^{16} bytes and a total main memory size of 2^{24} bytes. A process table is maintained that includes a pointer to a partition for each resident process. How many bits are required for the pointer?

4. [6 pt] What is the purpose of a translation lookaside buffer?

5. [9 pt] Suppose the page table for the process currently executing on the processor looks like the following. All numbers are decimal, everything is numbered starting from 0, and all addresses are memory byte addresses. The page size is 1024 bytes.

Virtual page number	Valid bit	Reference bit	Modify bit	Page frame number
0	1	1	0	4
1	1	1	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

   What physical address, if any, would each of the following virtual addresses correspond to? (Do not try to handle any page faults, if any.)

   (a) 1052

   (b) 2221

   (c) 5499

6. [12 pt] A minicomputer uses the first-fit system for memory management. Initially, it has one free block of 256MB at address 0. The requests come in the following order:

Arrival time	Burst time	Memory needed
0	19	219
0	15	81
1	14	117
3	18	241
6	5	14
7	3	152
11	4	200
12	4	106

How many free blocks are left at times 10, 20, 30, 40, 50, and 60, and what are their size and address?

7. [6 pt] How do you achieve security by obscurity in a Linux file system?

8. [6 pt] I have a floppy disk with 1.44MB [unformatted] capacity. The data blocks are 256 bytes each.

   (a) The OS keeps track of free space by using a bit vector approach. What is the size of the bit vector for this floppy? If the OS stores the bit vector on the floppy for recovery, how many blocks are left over to be used to store data.

   (b) Now, consider that the same floppy is formatted using UFS, with an empty boot block of size 1 block. Consider 1 block to be allocated for super block. Let each inode require 1024 bytes. What can be the maximum formatted capacity of the floppy? What is the maximum file size that can be stored on this floppy if the system uses 12 direct blocks, 1 single indirect block, and 1 double indirect block?