CS 4410 Operating Systems
Prelim II, Fall 2013
Profs. Sirer and van Renesse

Name:_____ NETID:_____

- **This is a closed book examination. It is 10 pages long. You have 120 minutes.**
- **No electronic devices of any kind are allowed.**
- **If you are taking this exam during the makeup period, you may not leave the exam room prior to the end of the exam, and you may not take an exam booklet with you.**
- **You may use Python, C, or low-level pseudo-code in answer to any coding question. Descriptions written mostly in English will get no credit as responses to coding questions. In addition to correctness, elegance and clarity are important criteria for coding questions.**
- **Assume Mesa semantics for all monitors. Show your work for partial credit. Brevity is key.**

[15] **1.**

a. Your program has just met an untimely end, halting mid-computation and producing only the text "segmentation fault" printed to the screen. What precisely does this mean?

b. A program calls malloc to request more memory from the operating system.
   i. what system call would malloc use to request more memory of the OS?

ii. outline the steps the OS takes in response to this system call

iii. The program now attempts to write into its freshly-allocated memory. What role does the OS and the hardware play in this operation? Outline all the steps that would be necessary to complete this write. Include the role of hardware (such as the TLB) if necessary.

[20] **2.**

After your previous successes at writing synchronization code, you have been selected to write robust software for the Emergency Room at the Systems Hospital. Specifically, the software will schedule and match incoming patients with doctors.

The hospital classifies patients into three categories: Critical, Sustainable and Minor. At Systems Hospital, all patients will wait in a single line and there are a few other rules the hospital follows:

1. All Critical patients must be treated before Sustainable and Minor patients
2. All Sustainable patients must be treated before Minor patients.
3. The order within patients of the same category must be maintained.

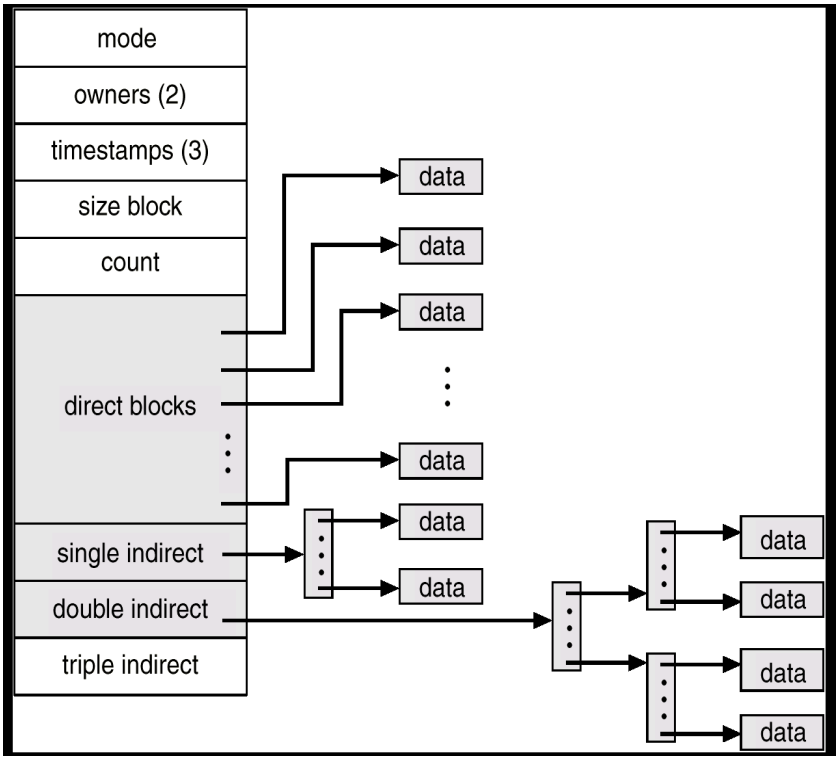
On a given day, there are 10 doctors total and all doctors are capable of treating patients in Sustainable and Minor categories. However, only some of the doctors are capable of treating patients in Critical category. All doctors must finish treating his/her current patient before treating the next patient. Also, no patient should be waiting in line if there are free doctors.

Fill in the methods for class SystemsHospital, using monitors and Mesa-style condition variables.

```
from __future__ import with_statement
from threading import Thread, Lock, Condition, Semaphore
import random

def delay():
    time.sleep(random.randint(0, 2))

# condition = ["Critical", "Sustainable", "Minor"]
class SystemsHospital:
        def __init__(self):




        def patient_enter(self, condition):




        def patient_exit(self, condition):
```

```
        def doctor_enter(self, can_handle_critical):









        def doctor_exit(self, can_handle_critical):









class Patient(Thread):
        def __init__(self, hospital, condition):
                Thread.__init__(self)
                self.hospital = hospital
                self.condition = condition

        def run(self):
                self.hospital.patient_enter(self.condition)
                delay()
                self.hospital.patient_exit(self.condition)

class Doctor(Thread):
        def __init__(self, hospital):
                Thread.__init__(self)
                self.hospital = hospital
                self.ability = random.randrange(0,2) == 0

        def run(self):
                while True:
                        self.hospital.doctor_enter(self.ability)
                        delay()
                        self.hospital.doctor_exit(self.ability)

hospital = SystemsHospital()
conditions = ["Critical", "Sustainable", "Minor"]
for i in range(100):
        c = condition[random.randrange(0, 3)]
        Patient(hospital, c).start()
for i in range(10):
         Doctor(hospital).start()
```

[20] **3**.
The above figure shows the i-node organization for a UNIX-filesystem-like file allocation scheme. The number of direct blocks is d, and each i-node contains one single, double, and triple indirect blocks, respectively. The size of each data block is b, and can contain at most n disk addresses.



(1) Calculate the maximum file size that this allocation scheme can support, in terms of d, b, n.

(2) Suppose that d=12, b is 1KB, and n=256. Further suppose that there is a file /tmp/data, of maximal size, and that all directory contents fit in a single block. Calculate how many disk accesses are required for a 4KB disk read from /tmp/data, starting at offset 268K, for a freshly booted operating system that has no blocks cached. Show your reasoning for full credit.

(3) List at least two ways in which the i-node organization shown above is a win over the way the FAT metadata is organized.

[10] **4**.

A port scanner is a tool that determines which ports are open on a target host. It does this by attempting to communicate with each port on the target host, and seeing whether the target host responds. For example, a simple port scanner might first try port 1, then port 2, and so on, all the way up to port 65535. Port scanners can operate on either TCP or UDP ports.

It is useful for port scanner results to be accurate--when the scanner reports that a port is open, it really is open, and when a port is closed, it really is closed.

Suppose you are writing a port scanner. Is it easier to get accurate results when scanning TCP ports, or when scanning UDP ports?

Give two reasons to explain why this is so.

[20] **5**.

Untold, Inc. is producing a new line a CPUs and wants to port Linux to it. On their first model, the Ydoit1, they have a 32-bit address space, 4 Kbyte pages, and a segmentation table with entries of the following form:

Start	Size	Writable

Here Start is the page number of the first page in the segment, Size the number of pages in the segment, and Writable a bit that specifies if the segment can be written. In addition, the Ydoit1 has a software-loaded TLB with 4 entries. The TLB is only used when the CPU is in user (not supervisor) mode, and uses a FIFO replacement strategy. The entries have the following form:

Virtual Page Number	Physical Page Number	Writable	Dirty

The Untold designers build a new development board with 1 Gigabyte of memory starting at physical address 0x00000000. They decide that Linux should be mapped in the top half of the virtual address space, leaving the bottom half (starting at physical address 0x00000000) to user programs.

An executable “a.out” has a “code” segment consisting of 4 pages starting at page 1, a “data” segment consisting of 3 pages, and a stack that can grow to 8 pages. The stack grows down.

a) Why does the code segment not start at page 0?

b) What should Linux port put in the segmentation table (use hexadecimal numbering):

Start	Size	Writable
.		
.		
.		
.		

The first user program to execute is “a.out”, and executes the following code (starting at address 0x1100), with the stack pointer pointing to the top of the stack segment:

```
0x1100:   push #0xA000          // push number 0xA000 onto stack
0x1104:   syscall SYS_SETBRK    // system call trap: set the top
                                // of heap to 0xA000.
0x1108:   …                     // rest of program
```

c) Describe the exceptions that will occur as this code is executed and how Linux will handle them.

d) Assuming that frames (physical pages) 0x100 to 0x200 are available for allocation to user processes and frames are allocated in order starting at frame 0x100, what does the TLB contain when “a.out” reaches 0x1108?

	Virtual	**Physical**	**Writable**	**Dirty**
0				
1				
2				
3				

Suppose that “a.out” now executes the following code:

```
0x1108:   move #1, 0x7010       // set memory location 0x7010 to 1
0x110C:   move 0x8020, R1       // R1 is a register
0x1110:   move R1, 0x9030
0x1114:   move R1, 0x6000
```

e) What’s in the TLB at the end of executing these instructions?

	Virtual	**Physical**	**Writable**	**Dirty**
0				
1				
2				
3				

[15] **6**.

As you know, TCP provides a reliable, ordered stream abstraction, while UDP provides an unreliable datagram abstraction. Currently, YouTube videos are delivered almost exclusively over TCP.

a. Describe the cons of using TCP to deliver video over the Internet.

b. Describe the pros of using TCP to deliver video over the Internet.

c. If you could design any transport protocol you like, describe the type kind of transport protocol you would implement to deliver video over the internet. You may assume that the video file formats are such that the clients can tolerate packet loss.