CSE 314: Quiz
February 19, 2022
Time: 40 Minutes

Student ID: _____
This quiz is worth a total of **45 points**.

Part A: Short Questions

Answer each of the following Short Questions

1. (a) In xv6, a page table is stored in physical memory as a two-level tree. Suppose, instead of a two-level page-table tree, for better granularity, we have three-level page-table tree structure for paging. Based on Figure 1, please find the values of m, n, p, k, i, and j. [5]

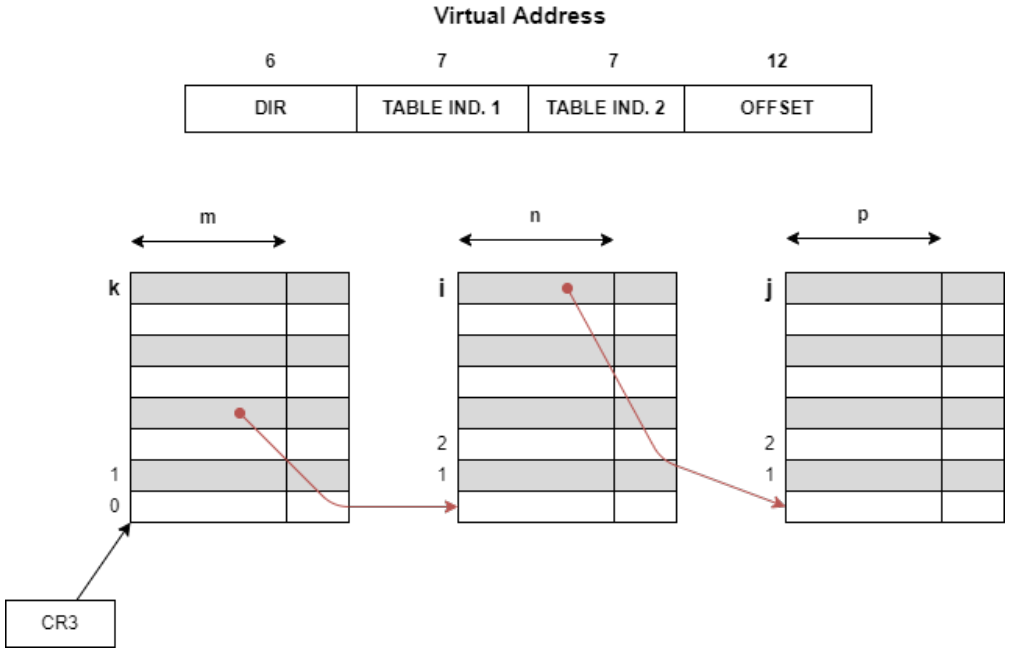

Figure 1: Modified page-table structure for xv6

(b) Suppose, the virtual address of a page is 0xE79EAA5B and the value of CR3 register is 0x59382453. Find the physical address of the required page. The virtual address and page-table should follow the structure as shown in Figure 1. Also, make and state necessary assumptions for finding the physical address. *You must show your work to receive full credit.*

2. Write a shell script program that will take two argument from the user. Your program needs to change the file extension from first argument to second argument to all applicable files in the current working directory. [5]
For example, the command *./sample.sh .dat .txt* will change the extension of all .dat files to .txt file in the current working directory.

3. Write shell command(s) to perform the following jobs. (a) Print the lines between 15 to 22 (both inclusive) from the beginning of a text file with more than 100 lines. (b) Report all usage of the command 'grep' from your terminal history to 'grep.txt' file. [5]

4. Consider a stairway with X number of steps. On every step, there can be at most one person moving in a particular direction, that is, on every step at a time there can be at most two persons of which one is moving upwards and the other moving downwards. No overtaking is allowed here. A person moves up/down to the next step only if that step is empty. Assume, each person stands for 10 ms at each step. Note that upto 2X number of persons are allowed to stay in the stairway simultaneously, where each person stands on a particular step facing a particular direction. [6]

   To model the above scenario in your program, you have been asked to create a thread per person. Now answer the following questions.

   (a) You have declared two counting semaphores: Sem_upward and Sem_downward. You have initialized each semaphore with the value X. When a person wants to go upward one step, Sem_upward is decremented once and is incremented when the step is completed. Similarly, when a person wants to go downward one step, Sem_downward is decremented once and is incremented when the step is completed.

   Explain how this implementation fails to model the given scenario.

   (b) To model the given scenario appropriately, how many semaphores will you require? Describe the type (i.e., binary/counting) and initial value of each semaphore you will declare. Also state the purpose/usage of the declared semaphores.

5. Suppose you have been given the task of adding a Translation Lookaside Buffer (TLB) on top of task 2 of your last assignment on xv6. Mention the files and the necessary changes in the respective files for adding this functionality (only pseudocode will suffice). [8]

6. C1S7E is a graduate-level student at B0U1E7T. They loves minimalistic design of systems. So they decided to get their hands dirty to polish xv6 & get rid of unnecessary chunks! He carefully inspected the codes of xv6 scheduler: [3]

```
void scheduler(void)
{
  struct proc *p;
  struct cpu *c = mycpu();
  c->proc = 0;

  for(;;){
    sti();
    acquire(&ptable.lock);
    for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
      if(p->state != RUNNABLE)
        continue;
      c->proc = p;
      switchuvm(p);
      p->state = RUNNING;
      swtch(&(c->scheduler), p->context);
      switchkvm();
      c->proc = 0;
    }
    release(&ptable.lock);
  }
}
```

They asked his friends C1M7U about this function *sti*() used here & got to know this is here to make sure **interrupts** are enabled. C1S7E decided to get rid of that *sti*() function and recompiled his xv6 again. Voila! It works! Was that a good idea to omit? State your reasons. You have to show appropriate scenario to get credits for this question.

7. Look at the following code of *wait*() system call from **proc.c**. Note that this is an simplified version of the original xv6 code with lots of nitty-gritty details redacted: [4]

```
wait(void) {
    for(;;){
        if there is an exited child {
            clean it up ...;
            return pid; // return the child pid
        }

        if(this process has no children || proc->killed){
            ...
            return -1; // error
        }

        // wait for a child to exit.
        sleep(proc, ...);
    }
}
```

C1S7E decided to rewrite the system call as follows:

```
wait(void) {
    for(;;){
        if(this process has no children || proc->killed){
            ...
            return -1; // error
        }

        // wait for a child to exit.
        sleep(proc, ...);

        if there is an exited child {
            clean it up ...;
            return pid; // return the child pid
        }
    }
}
```

Can anything go wrong as a result of the simple modification? **[Hint: Think when can** *wait*() **act incorrectly, if that does at all.]**

Part B: Multiple Choice Questions

Answer each of the following Multiple Choice Questions.

1. From where the read statements would read if the following statements are executed in a shell script code? [1]

```
exec < file 1
exec < file 2
exec < file 3
read line
```

(a) It would read all the files
(b) It would not read any files
(c) It would read all the files in reverse order
(d) It would read only file 3

2. What is the output of the following code? [1]

```
os=linux
echo 1.$os 2."$os" 3.'$os' 4.os
```

(a) 1.linux 2.linux 3.linux 4.linux (b) 1.linux 2.linux 3.$os 4.linux
(c) 1.linux 2.linux 3.$os 4.os (d) 1.linux 2.$os 3.$os 4.$os

3. Which of the following APIs can be used for synchronization between threads? [1]
(a) pthread_exit (b) pthread_cancel
(c) pthread_join (d) pthread_self

4. To ensure difficulties do not arise in the original readers – writers problem, ____ are given exclusive access to the shared object. [1]
(a) readers (b) writers
(c) readers and writers (d) none of the mentioned

5. Carefully look at the following POSIX C code snippet: [2]

```
int fl = open("xv6-mm-easy-prob.dat");
char *ptr = (char *) mmap(0, 131072, PROT_READ | PROT_WRITE, MAP_SHARED);
uint sum = 0;

for (int k = 0; k < 3; k++){
    for (int i = 0; i < 8192; i += 1) {
        sum += ptr[i];
    }
}
uint val1 = 0;
uint val2 = 0;
for (int j = 20472; j <= 45056; k++){
    sum += ptr[i]
    val1 = prt[1]
    val2 = ptr[45150]
}

printf("%d\n%d\n%d\n", sum, val1, val2);
```

Assuming standard xv6 pages and a FIFO page-replacement policy, how many page-faults will occur for the above code? (MAX_PSYC_PAGES = 4).
(a) 32 (b) 9
(c) 11 (d) 13

6. In which case panic() function is called in xv6? [1]
(a) Spurious interrupt (b) Bug in user program
(c) Kernel bug (d) Invalid system call number

7. Which of the following is false in xv6? [1]
(a) Value of PTXSHIFT is 12 (b) Each process has a separate page directory
(c) All the pages of a process are allocated in memory when the process starts (d) Both a and b

8. When is **trapret** run? [1]
(a) During swtch between two existing processes. (b) During process creation
(c) During allocating new process; before **forkret**. (d) During allocating new process; after **forkret**.