**CS 354 Fall 2025**

**Homework 1**

**Juan Florez - 039614388**

1. Suppose someone tells you that a Xinu system has three processes as follows:

Could the information be correct? Why or why not?

No, the information can’t be correct, in Xinu the scheduler always selects the highest priority process from the ready list to run. But here the process Y has a priority of 30 this is higher than process X that is 19, but it’s in READY state while the lower priority X is CURRENT, this state is impossible because of the rescheduling would have occurred to run Y instead.

2. Suppose Xinu system has six processes as follows:

(a) What will the ready list contain? (Indicate the head and tail of the list.)

head A-D-B-main tail

(b) After a timeslice event occurs, which process will be running? Explain.

The process A will be moved to READY and enqueued at the end of its priority group, the scheduler will then select the highest priority READY process, which is A.

3. Suppose Xinu system has six processes as follows:

If a timeslice event occurs, which processes, if any, will change state, what will the new states be, and what will the ready list contain?

The process A will change to READY and be enqueuedat the end of the priority 20 group, the process B will become CURRENT. The ready list will contain head A tail, since A and B both had priority 20 and A is queueed after B but B is now running.

4. Suppose Xinu system has six processes as follows:

If process A calls resume on process C, which processes, if any, will change state, what will the new states be, and what will the ready list contain? Explain.

The process C will change to READY because resume is called on it, but resume is intended for SUSPENDED processes, calling resume on a non SUSPENDED process returns SYSERR and has no effect. The ready list remains head B tail and process A remains CURRENT.

5. Suppose Xinu system has five processes as follows:

If process A calls resume on process E, which processes, if any, will change state, whatwill the new states be, and what will the ready list contain? Explain.

The process E will change to READY and be enqueued on the ready list, since its priority 20 matches A, rescheduling occurs, but A continues as CURRENT because equal priority processes don’t preempt. The ready list will contain head B – E tail.

6. Suppose Xinu system has five processes as follows:

If process A signals semaphore 13, which processes, if any, will change state, what will the new states be, and what will the ready list contain?

The process C will change to READY because signal increments the semaphore count, dequeuing and readying the first waiting process. The ready list will contain head B – C tail, all priority 20. Process A remains CURRENT.

7. Suppose Xinu system has five processes as follows:

(a) Suppose process A creates and resumes a new process, F, that has priority 21. Which processes, if any, will change state, what will the new states be, and what will the ready list contain?

The new process F is created and resumed so it becomes READY, since F priority 21 is higher than A 20, rescheduling occurs. The ready list will contain head A – B tail and C, D, E unchendged.

(b) Suppose that when process F runs, it resumes D, then resumes E, and then signals semaphore 13. Finally F exits. What will the states of the processes be, and what will the ready list contain?

When F runs the resuming D makes D READY, resuming E makes E READY, signal semaphore 13 makes C READY, rescheduling after each: After resuming D and E, F continues then after signal, C makes READY, F continues then exit and the highest ready is A-B-C-D-E all 20. The ready list will be head A – B – D – E – C tail.

8. A scientist uses Xinu to create and resume 100 processes with priority 10. After all 100 processes have been resumed, the main process exits, and there are no other processes running. Each of the processes starts with an input value, performs a complex computation that requires several seconds of computation, and then uses kprintf to print a single floating point value. After printing its value the process exits.

Let's refer to the processes as P1 through P100, and assume they are created and resumedin order from 1 to 100.

(a) After all 100 processes have been resumed and just before the main process exits, will the ready list have the most recently created process at the head (i.e., will the ready list contain P100, P99, P98, and so on)? Explain.

No, the ready list will not have the most recently created at the head, the ready list is ordered strictly by priority, all 100 processes have the same priority 10, so they are enqueued in the order resumed. The ready list will contain p1 at head then p2 …. P100 at tail.

(b) Will the 100 output values be in random order or will they be ordered by the processes? That is, will the processes finish randomly or will they have definite order, such as P1, P2, P3... or perhaps P100, P99, P98,...?

The output values will be ordered by the processes p1, p2, p3… p100, with equal priorities and no other processes, also the timeslicing will cycle through them in ready list order round robin within priority.

9. You have been asked to debug someone's implementation of wait and signal. You insert kprintf statements to dump the semaphore table.

(a) You find that the semaphore with ID 3 has a count of 4 instead of 3. Does that indicate an error in the code? Explain why or why not.

No, it does not indicate an error, A semaphore count of 4, could result from extra signals without waits which is valid if no processes are waiting.

(b) You find that the semaphore with ID 17 has a count of -7. What must be present onthe list of waiting processes if the implementation is correct? Explain.

A count of -7 means 7 processes must be on the waiting list, the semaphores work with negative count -n, this indicates exactly n processes waiting.

(c) You find that the semaphore with ID 2 has a count of 12, but only has 11 processes on the list of waiting processes. Does that indicate an error in the code? Explain why or why not.

Yes, it indicates an error. A count of 12 means no processes should be waiting but 11 on the wait list contradicts this positive count implies empty wait queue.

10. The Xinu code returns SYSERR if someone attempts to create a semaphore with an initial value of -1. Explain why that check is in the code.

The check prevents invalid semaphores, initial count must be non negative and negative would imply waiting processes at creation, which is illogical and could corrupt the wait queue.

11. The Xinu implementation of suspend does not place a suspended process on any list. Imagine changing the suspend code to place a suspended process on a list of all suspended processes. Also imagine changing the code in resume to search the list of suspended processes to determine whether the argument specified a process that was indeed suspended. Would such an implementation run faster than the current implementation? Explain.

No, it would run slower, current suspend marks the process state as SUSPENDED, resume checks state and readies if SUSPENDED is the same, this increas overhead without benefit.

12. Suppose that someone calls

create(fun1, 4096, 99, "newpr" , 2, 3, 4)

(a) How large will the stack of the new process be?

The stack will be 4096 bytes.

(b) Where will the process begin executing?

The process will begin executing at function fun1.

(c) How many arguments will be passed to the top-level function?

2, (3 and 4).