



Honor Code of the School of Engineering

"All students taking courses in the School of Engineering agree, individually and collectively, that they will not give or receive unpermitted aid in examinations or other course work that is to be used by the instructor as the basis of grading."

-From the Graduate/Undergraduate Bulletin

I have read, understood, and agree to abide by the Honor Code of the School of Engineering.

Name: _____

ID: _____

Signature: _____

Date: _____

1. [10 points]
2. [40 points]
3. [50 points]
4. [20 points]
5. [20 points]
6. [20 points]
7. [20 points]
8. [20 points]

Total Score:

Midterm Examination

**CSEN 283 Operating Systems
Department of Computer Science and Engineering
Santa Clara University**

Dr. Ming-Hwa Wang
Phone: (408) 805-4175
Course website:
Office Hours:

Fall Quarter 2024
Email address: m1wang@scu.edu
<http://www.cse.scu.edu/~m1wang/os/>
Tuesday & Thursday 9:00-9:30pm, Saturday 11:00am-12:00pm

1. [10 points] Please give the whole names for the following acronyms: SJF, LDT, PCI, CTSS, PSW.
2. [40 points] True or false problems with wrong-answer penalties:
 - a) Suppose that a 32-bit virtual address is broken up into four fields, a, b, c, and d. The first three are used for a three-level page table system. The fourth field, d, is the offset. Does the number of pages depend on the sizes of all four fields?
 - b) Timesharing systems generally use non-preemptive processor scheduling.
 - c) Pipes are an example of a blocking communication mechanism.
 - d) Linux is a microkernel OS.
 - e) A lower priority job can block a high priority one (e.g., priority inversion problem) even a priority scheduling is used.
 - f) An unsafe state is one in which there is at least one sequence of resource allocations to processes that does not result in a deadlock.
 - g) Does the busy waiting solution using the turn variable work when the two processes are running on a shared-memory processor, that is, two CPUs sharing a common memory?
 - h) Algorithms for mutual exclusion and deadlock must depend on access to common memory.
3. [50 points] Simple questions:
 - a) Consider a computer system that has cache memory, main memory (RAM) and disk, and the operating system uses virtual memory. It takes 2 nsec to access a word from the cache, 10 nsec to access a word from the RAM, and 10 ms to access a word from the disk. IF the cache hit rate is 95% and main memory hit rate (after a cache miss) is 99%, what is the average time to access a word?
 - b) A computer has 4 GB of RAM of which the operating system occupies 512 MB. The processes are all 256 MB (for simplicity) and have the same

characteristics. If the goal is 99% CPU utilization, what is the maximum I/O wait that can be tolerated?

- c) The Banker's algorithm is being run in a system with m resources classes and n processes. In the limit of large m and n , the number of operations that must be performed to check a state for safety is proportional to $m^a n^b$. What are the values of a and b ?
- d) It has been observed that the number of instructions executed between page faults is directly proportional to the number of page frames allocated to a program. If the available memory is doubled, the mean interval between page faults is also doubled. Suppose that a normal instruction takes $1 \mu\text{sec}$, but if a page fault occurs, it takes $2001 \mu\text{sec}$ (i.e., 2 msec) to handle the fault. If a program takes 60 sec to run, during which time it gets 15,000 page faults, how long would it take to run if twice as much memory were available?
- e) A real-time system needs to handle two voice calls that each run every 6 msec and consume 1 msec of CPU time per burst, plus one video at 25 frames/sec, which each frame requiring 20 msec of CPU time. Is this system schedulable? Why?

4. [20 points] You are given the following data about a virtual memory system:

- The TLB can hold 1024 entries and can be accessed in 1 clock cycle (1 nsec).
- A page table entry can be found in 100 clock cycles or 100 nsec.
- The average page replacement time is 6 msec.

If page references are handled by TLB 99% of the time, and only 0.01% lead to a page fault, what is the effective address-translation time?

5. [20 points] Does the code below provide mutual exclusion? If yes, please explain why. If not, show an interleaving in which mutual exclusion is not preserved.

1 System:

```
int turn = 1;
boolean t1WantsToEnter = false;
boolean t2WantsToEnter = false;
startThreads(); // initialize and launch both threads
```

Thread T1:

```
void main() {
    while ( !done ) {
        t1WantsToEnter = true;
        while ( turn != 1 ) {
            while ( t2WantsToEnter );
            turn = 1;
        } // end while
        // critical section code
        t1WantsToEnter = false;
```

```
// code outside critical section
} // end outer while
} // end Thread T1
Thread T2:
void main() {
    while ( !done ) {
        t2WantsToEnter = true;
        while ( turn != 2 ) {
            while ( t1WantsToEnter );
            turn = 2;
        } // end while
        // critical section code
        t2WantsToEnter = false;
        // code outside critical section
    } // end outer while
} // end Thread T2
```

6. [20 points] A system that uses Banker's-Algorithm deadlock avoidance has five processes (1, 2, 3, 4, and 5) and uses resources of four different types (A, B, C, and D). There are multiple resources of each type.

a) Is the state of the system below safe?

Process	Current Loan				Maximum Need				Current Claim			
	A	B	C	D	A	B	C	D	A	B	C	D
1	1	0	2	0	3	2	4	2	2	2	2	2
2	0	3	1	2	3	5	1	2	3	2	0	0
3	2	4	5	1	2	7	7	5	0	3	2	4
4	3	0	0	6	5	5	0	8	2	5	0	2
5	4	2	1	3	6	2	1	4	2	0	0	1

Total Resources Resources Available

A	B	C	D	A	B	C	D
13	13	9	13	3	4	0	1

b) Explain your answer. If the system is safe, show how all the processes could complete their execution successfully. If the system is unsafe, show how deadlock might occur.

7. [20 points] Using the program below, identify the values of pid at lines A, B, C, and D. (Assume that the actual pids of the parent and child are 2600 and 2603, respectively.)

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
int main ( ) {
    /* fork a child process */
    int pid = fork( );
    if ( pid < 0 ) { /* error occurred */
```

```

    fprintf(stderr, "Fork Failed\n");
    return 1;
} else if ( pid == 0 ) { /* child process */
    int pid1 = getpid( );
    printf("child: pid = %d\n", pid); /* A */
    printf("child: pid1 = %d\n", pid1); /* B */
} else { /* parent process */
    int pid1 = getpid( );
    printf("parent: pid = %d\n", pid); /* C */
    printf("parent: pid1 = %d\n", pid1); /* D */
    wait(NULL);
}
return 0;
}

```

8. [20 points] Suppose that the following processes arrive for execution at the times indicated: P_1 arrives at time 0.0 with burst time 8, P_2 arrives at time 0.4 with burst time 4, and P_3 arrives at time 1.0 with burst time 1. Each process will run for the amount of time listed. In answering the questions, use non-preemptive scheduling, and base all decisions on the information you have at the time the decision must be made.
- What is the average turnaround time for these processes with the FCFS scheduling algorithm?
 - What is the average turnaround time for these processes with the SJF scheduling algorithm?
 - The SJF algorithm is supposed to improve performance, but notice that we chose to run process P_1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes P_1 and P_2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling.