

Exam #1

CSE 3320.002

Spring 2015

Name: \_\_\_\_\_

UTA ID: \_\_\_\_\_

“I certify that the following work is my work alone and I will follow the highest standards of integrity and uphold the spirit of the Honor Code”

Signature: \_\_\_\_\_

Directions: This is a closed book, closed notes exam. You may use a hand written 3x5 note card with notes. Please answer the questions briefly. Complete sentences are not necessary. Write your answers legibly. Unreadable answers will be counted wrong. You may write on back if needed.

1. [8 pts] How do system calls vary from function calls in terms of execution? Why do they vary?

2. [8 pts] Explain the difference between cooperative and pre-emptive multitasking. What is an advantage of each?

3. [8 pts] For the following code, what will be printed? For return values from fork use any integer that makes sense from a system standpoint:

```
pid_t pid_1;
pid_t pid_2;
int status;

pid_1 = fork();

if (!pid_1)
{
    wait(&status);
    printf("Tatooine\n" );
    if (!(pid_2 = fork()))
    {
        wait(&status);
        printf("Kashyyk\n" );
    }
    else
    {
        wait(&status);
        printf("Hoth\n" );
    }
    exit( 0 );
}
wait(&status);
printf( "Dagobah\n" );
exit( 0 );
```

4. [8 pts] For the following code, describe what happens, in what order, and what will get printed (in what order). For return values from fork use any integer that makes sense from a system standpoint:

```
execl( "/bin/mkdir", "/bin/mkdir", "GoT", NULL );
child_pid = fork();
if (child_pid == 0)
{
    printf("Stark\n")
}
else
{
    int status;
    int child_pid = fork();
    if( child_pid == 0 )
    {
        int child_pid = fork();
        if( child_pid == 0 )
        {
            printf("Arryn\n");
        }
        else {
            wait( &status );
            printf("Baratheon\n")
        }
    }
    else {
        int child_pid = fork();
        if( child_pid == 0 )
        {
            printf("Lannister\n");
        }
        else {
            wait( &status );
            printf("Greyjoy\n");
        }
    }
    wait( &status );
    printf("Targaryen\n");
    exit( 0 );
}
```

5. [6pts] For a CP/M disk, the disk had 40 tracks of 32 sectors each sector was 512 bytes long and the disk had two sides. The disk had the “standard” CP/M disk directory of 64 entries. Standard CP/M block sizes are 128 bytes. You do not need to calculate the final result. Just show the calculation and reasoning.

- A. What is the raw disk capacity?
- B. For this disk, how many 1 byte files could be stored?
- C. What is the largest file size that could be stored?

6. [6pts] What data structure does the kernel use to hold the state of the CPU and of the process? How many of this structure are there? Name three elements in the structure.

7. [6pts] Describe external fragmentation. What technique did PALM OS use to correct memory fragmentation? How did they avoid causing problems for running programs?

8. [50 pts]

The following is a list of processes

Process ID	Arrival Time	Runtime (seconds)	Priority
1	0	3	2
2	0	2	1
3	4	5	3
4	6	4	4
5	7	2	2
6	8	1	3

Lower priority values are higher priority. When arrival time is the same, select lowest process number first, if more than one process can be selected use "most fair" - allow next (newest) process to run next.

1. Please show the GANTT chart for a FCFS scheduler.
  1. What is the turnaround time for Job 2 and Job 5?



Process ID	Arrival Time	Runtime (seconds)	Priority
1	0	3	2
2	0	2	1
3	4	5	3
4	6	4	4
5	7	2	2
6	8	1	3

2. Please show the GANTT chart for a Shortest Job Next (SJN) scheduler (with no preemption).
  1. What is the response time for Job 1 and Job 3?

Same table:

Process ID	Arrival Time	Runtime (seconds)	Priority
1	0	3	2
2	0	2	1
3	4	5	3
4	6	4	4
5	7	2	2
6	8	1	3

3. Please show the GANTT chart for a SJN scheduler, with preemption. Show average turnaround time, average response time, average wait time.

Bonus Question (5pts a piece)

B1. How many context switches occurred in problem 8.3 SJN with preemption?

B2. Draw and describe to x86 protection rings.