Q2 - 32 points

I - UNIX Environment

1. What does the BIN directory contain? What does the DEV directory contain?

**BIN: The binary directory: contains executable files and most Unix commands.**

**DEV: Device directory: It contains special or device files.**

1. Discuss the creation of a process from the point of view of ADDRESS SPACE.
2. **The child is a duplicate of the parent (same address space)**
3. **The child is a separate program (different address space)**
4. What is the name of the system call (command) that can overwrite the default address?

**Exec(…)**

1. What happens to a child process if its parent is terminated?

**The child becomes an orphan process and is adopted by the init process.**

1. What are the return values for the FORK() command, and what are their meanings?

**The fork command will return a negative number if the creation of the new process is unsuccessful.**

**Return 0 for the new child process.**

**Returns a positive value which is the PID of the successfully created process.**

1. Define a ZOMBIE process.

**Zombie process is a process that has been terminated but remains in memory. This happens when the child process terminates before the parent process.**

II -

1. Explain the main difference between a MODE SWITCH and a FULL CONTEXT SWITCH. Give examples of each type of switch.

**Mode Switch – Saves the context of the processor, then sets the PC to the starting address of an OS, the interrupt handler routine. Then it switches from user mode to system mode.**

**Example: I/O Operations**

**Full Context Switch – Saves the context, Updates the PCB of process in Run state, Move to a queue. Then Selects another process, updates PCB, update memory management, restore the context of the processor.**

**Example: Wait(), exit().**

1. **II (H) Test and Set**

p[i] (Lock is initially set to false)

while(true){

while(TS(LOCK)){}

CS

LOCK=FALSE;

}

Assume Process 1 is in the CS. For Process 1 to be in the CS, Lock must be set to true.

If Process 2 tries to enter the CS, Process 2 will busy wait. For Process 2 to exit BW, TS (LOCK) must return false.

For TS (Lock) to return false, Process 1 must finish the CS, which will set TS (LOCK) to false.

Since this is an atomic implementation, no Process can get into the CS while another process is checking the LOCK. The Process cannot be interrupted.

**What if it was non-atomic?**

If it was non-atomic, the process can be interrupted during the CS, however, Lock will remain the same since the lock is only changed after the CS, which does not allow another process to enter CS.

STORY QUESTION -

Children wait to get their gift. Each child process gets its turn[i] by computing number[i]. A clown has in his hat ???? each ball has a different number, from 1 to 10. After all children have their turn set, the clown is ???? -are the gifts (there are 10 gifts). The clown picks a random ball, and gives the gift to the child whose number is the same with the number of the ball (after that, he throws away the selected ball).

Variables:

turn[] = 0

number[] = 0

called[] = 0

ballsNumber = 10

i = 1,.......10 (N=10)

child(int i)

{

number[i] = 1+max(number[1],number[2],....number[N});

turnpi] = number[i];

while(!called[i]) {busyWait;}

getTheGift(); //sleep

ballsNumber--;

goHome(); //sleep

}

clown()

{

while(ballsNumber>0)

{

pickABall(); //sleep

for(int j=1; j<=N; j++)

{

if(turn[i]==numberOnBall)

{

called[j]=true;

giveTheGift(); //sleep

}//if

}//for

}//while

leave;

}//clown

\*ALL CHILD PROCESSES EXECUTE CONCURRENTLY.

(A) Is it possible for 2 children to compute the same value for number[i]? Explain why and give the execution sequence that can show it.

(B) Consider that 6 children already computed their number[i]. Give an execution sequence by which after these 6 children computed their number, the largest computed number[i] = 4.

(C) On the hypothesis that each child has a different number[i], is it possible for children to compete for the same give (because turn[i] values are the same)? Explain. If yes, give the execution sequence that will show it.

(D) Consider that at this point, all cihldren have their turn[i] updated (are done with the execution of turn[i]=number[i];) Is it possible for a child to starve (never be called), by busy waiting in the while loop? Explain. If yes, give the execution sequence that will show it.

(E) Under the hypothesis that each child had a different turn[i] value and received a gift, is it possible for the clown to not be able to go home because the while condition is still true? Give the sequence that will show this situation.

Extra Credit - 6 points

(Giving the execution sequence) that if TS is not executed automatically, the Mutual Exclusion condition with TS fails to satisfy the condition. Show where the interrupt should occur.