Fall 2020 CSC 332 Quiz 3 300 Points 90 minutes

Each question is 100 Points.

Q1. Consider the following code.

What are **all possibilities about output** (**including the possibility of no output**), and show **one** scenario to get **each output possibility**.

int i=0; //in heap

cobegin

S11: if (i == 0) S21: if (i == 0)

S12: i= i+1; S22: i= i - 2;

coend;

print(i);

**Answer:**

1: s11,s12,s21, exit both

-2: s21,s22,s11, exit both

-1: s11,s21,s12,s22

Q2. Consider the following code. What is the **MAXIMUM** value printed? Give a **Scenario** for this value.

**NOTE: the scenario is critical. If the scenario is grossly wrong, then your answer is essentially a guess and you lose a lot of points.**

int counter=0; //in heap.

**cobegin**

for (int i=0; i<1000; i++) for (int i=0; i<1000; i++)

counter++; counter = counter - 1; //WATCH OUT!!

**coend**

print(counter);

**Answer:**

Max=1000

**Scenario:**

**Thread 1 Thread 2**

As part of counter++, Load value of

counter into R1.

Now i==counter==R1==0

Do many iterations, exit.

R1++

Store R1 into counter.

Increment i

Now counter=i==1

(counter++; i++) 999 times.

Now counter==i==1000

Exit

Parent thread prints 1000

Q3.

1. Suppose two threads T1 and T2 are running concurrently in the same process **with a single CPU**. Suppose T1 does “Load X into register R1” machine instruction. Then the CPU scheduler switches the CPU to T2. Will T2 see the same vaue of R1 as loaded here by T1? (YES or NO)? Explain in less than 50 words. Be specific.

**Answer:**

**No.**

When we switch from T1 to T2, value of register R1 (in T1) is saved in memory and old saved value of R1 in T2 is loaded into the register R1. So T2 will see R1 value which was saved a while back.

1. Suppose two threads T1 and T2 are running concurrently in the same process **with a single CPU**. Is it possible to have a moment (time instant) where both threads have made a request to Memory Unit to read some data (ex. as part of a “Load … into register …” instruction and Memory Unit has not yet completed either of these two requests? Explain in less than 50 words.

**Answer:**

**No.**

All hardware interrupts take place at the end of current instruction execution (in the interrupt cycle). This includes hardware time interrupt.

So two threads cannot be in the middle of their instruction execution at the same time.

In the situation described in the question, the two threads are in the middle of their instruction execution. But this is not possible.

**Note:** This has nothing to do with simultaneous access in assumption A1. A1 is talking about NET EFFECT if simultaneous access takes place.

This question is about whether simultaneous access is even possible in hardware.