LE18.2

LE18.2.1: SVC Behavior

0.0/1.0 point (ungraded)

A Beta running with the OS described in lecture is running two processes:

```
        // Process 0:                  // Process 1:
        P0: GetKey()                   P1: ADDC(R0, 1, R0)
            WrCh()
            BR(P0)                         BR(P1)
```

GetKey() and WrCh() behave as follows:

- GetKey() -- returns the next ASCII character from the keyboard in R0; this SVC ensures that the next instruction following the GetKey() call is not executed until there is a character available.
- WrCh() -- writes the ASCII character found in R0 to the console.

You type a few characters and see them echoed.

1. What can you say about the value in the XP register when the instruction following the **GetKey()** SVC is executed?

   The XP register may contain different values on different iterations:

   False Answer: False

   It always has a high-order 0 bit:

   True Answer: True

   It is always a multiple of 4:

   True Answer: True

   It always has the value P0 + 4:

   True Answer: True

   Explanation
   Since the GetKey() supervisor call does not return to the user until there is a character available, that means that when it returns it is ready to execute the instruction immediately after the call to GetKey(). The address of this instruction is P0 + 4. All instruction addresses are a multiple of 4. Since we are back in user mode, the high-order bit of the PC is 0.

   You notice that Process 1 increments R0, whose value increases at some fairly constant rate **R** increments/second. You experiment with several changes below, not typing but monitoring the rate at which the count in Process 1's R0 increases. For each of the experiments below, you are asked to estimate its effect on the R0 counting rate, relative to **R**; choose among

   - **+LOTS:** the rate increases considerably, e.g., twice **R**.
   - **SAME:** the rate is about the same as **R**.
   - **-LOTS:** the rate is much lower, e.g., **R/2** or lower.

2. As an experiment, you eliminate Process 0 (so that only Process 1 is running). How does the rate of increase of Process 1's R0 change? Assume the scheduler time slice for each process is long compared to one iteration of either loop.

   How does the rate of increase of Process 1's R0 change?

   SAME Answer: SAME

   Explanation

Since process 0 spends most of its time waiting for a key, it very quickly gives up the rest of its time slice when there is no key press. So even when process 0 is running, process 1 has most of the actual running time allocated to it. So removing process 0 does not change the rate of increase of process 1's R0 very much.

You restore both processes and **eliminate the Scheduler() call** invoked by the GetKey() SVC when no character is ready to be returned.

3. How does this effect the rate at which R0 increases, relative to the original counting rate **R**?

-LOTS Answer: -LOTS

Explanation
If Scheduler() is not called from within the GetKey() SVC, then the rest of the time slice that was alloted to process 0 is just wasted. As a result, process 1 is run less frequently than it would be if the Scheduler() call was still there. The result of this is that R0 doesn't grow as rapidly.

4. Again running both Process 0 and Process 1, you now replace the single Scheduler() call invoked by GetKey() by **two consecutive** Scheduler() calls. How does the rate at which Process 1 counts change, relative to the original rate **R**?

-LOTS Answer: -LOTS

Explanation
If Scheduler() is called twice, then process 1 will be skipped so R0 will not get incremented.

Submit

Discussion

Topic: 18. Devices and Interrupts / LE18.2

Hide Discussion

Add a Post

Show all posts by recent activity

? Does key strike creates new process? 8
On slide #5, it shows that key strike triggers hardware asserting IRQ to request interrupt, which is not part of any existing current pr...

? question D: rate should not change 2
Doesn't the scheduler switch the process immediately? According to previous lecture, the schedule's function is to change the con...

‹ Previous Next ›

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap
Cookie Policy
Your Privacy Choices

Connect

Idea Hub
Contact Us
Help Center
Security
Media Kit

GET IT ON Google play
Download on the App Store

ICP 17044299 -2