Quiz #4: May 6, 2016

<table>
<thead>
<tr>
<th>Name</th>
<th>Athena login name</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jonathan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rakesh</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lucy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robert</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please enter your name and Athena login name in the spaces above. Enter your answers in the spaces provided after each question. You can use the extra white space and the backs of the pages for scratch work. We can give partial credit only if your work is shown on the exam.

Problem 1. Virtual Memory (10 points)

Consider a Beta processor that includes a 32-bit virtual address, an MMU that supports 4096 (2^{12}) bytes per page, 2^{24} bytes of physical memory, and a large Flash memory that serves as a disk. The MMU and the page fault handler implement an LRU replacement strategy.

(A) (2 Points) The designers are thinking about implementing the page map using a separate SRAM memory with L entries, where each entry has B bits. If the page map includes the standard dirty and resident bits, what are the appropriate values for the parameters L and B?

\[
\begin{align*}
\frac{2^{32}}{2^{12}} &= 2^{20} \text{ virtual pages} \\
\frac{2^{24}}{2^{12}} &= 2^{12} \text{ physical pages}
\end{align*}
\]

Appropriate value for the parameter L: \(2^{20}\)

Appropriate value for the parameter B: \(12 + 2 = 14\)

(B) (3 Points) If the designers decide to decrease the page size to 2048 (2^{11}) bytes but keep the same size virtual and physical addresses, what affect will the change have on the following architectural parameters? Use a letter “a” through “e” to indicate how the new value of the parameter compares to the old value of the parameter:

(a) doubled (b) increased by 1 (c) stays the same (d) decreased by 1 (e) halved

\[
\begin{align*}
\frac{2^{32}}{2^{11}} &= 2^{21} \text{ virtual pages} \\
\frac{2^{24}}{2^{11}} &= 2^{13} \text{ physical pages}
\end{align*}
\]

Size of page map entry in bits: \(B\)

Number of entries in the page map: \(A\)

Maximum percentage of virtual memory that can be resident at any given time: \(C\)

\[
\frac{2^{13}}{2^{20}} = 2^{-8} \quad \text{old} \quad \frac{2^{21}}{2^{13}} = 2^{8} \quad \text{new}
\]
(D) (4 points) A test program has been running on the Beta with a page size of \(2^{12}\) bytes and has been halted just before execution of the following instruction at location 0x1234:

\[
\text{ST(R1, 0x34C8, R31) } \mid \text{ PC = 0x1234}
\]

<table>
<thead>
<tr>
<th>VPN</th>
<th>D</th>
<th>R</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0x1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0x0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0x6</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>0x7</td>
</tr>
</tbody>
</table>

The first 8 locations of the page table at the time execution was halted are shown below; the least-recently-used page ("LRU") and next least-recently-used page ("next LRU") are as indicated. Assume that all the pages in physical memory are in use. Execution resumes and the ST instruction is executed.

Please show the contents of the page table after the ST instruction has completed execution by crossing out any values that changed and writing in their new values. Note that the D and PPN fields for a non-resident page do not need to be specified.

(E) (1 point) Which physical pages, if any, need to be written to disk during the execution of the ST instruction in part (D)?

Physical page numbers written to disk or NONE:

\[
\text{write } 0x34C8 \rightarrow \text{VPN 3} \rightarrow \text{miss} \rightarrow \text{write dirty LRU page to disk} \rightarrow \text{reuse PPN 2 for VPN 3} \rightarrow \text{ST set D bit.}
\]

Problem 2. Operating Systems (4 points)

Please implement the SNOOZE supervisor call handler for the Lab 7 operating system. To use SNOOZE, the user program first puts an integer time-of-day into R0, then executes the SNOOZE() SVC. The next instruction in the user program should be executed only when the time-of-day has reached (or passed) the time indicated in R0. When the user program resumes, the value in R0 should be 0. In the operating system, the time-of-day is maintained by an interrupt handler invoked by an IRQ signal asserted once per second. Your code should allow good CPU utilization by other running programs. Note that the user's register "j" can be accessed at User.Regs[j]. Please add the appropriate lines of code to the SNOOZE handler below:

```c
SNOOZE_h() {
    if (GetTimeOfDay() >= User.Regs[0]) {
        // set j to 0
        User.Regs[j] = 0;
    } else {
        schedule(); // after other procs get a turn.
    }
}
```
Problem 3. Interrupts and Real-time (7 points)

A particular real-time system has three interrupt handlers. The following table shows the maximum rate at which each interrupt occurs (rate), the time taken to execute each handler (service time), and the maximum allowable interval between the interrupt and completion of the handler (deadline). In your analysis, assume that A, B, and C interrupts can arrive at any time.

<table>
<thead>
<tr>
<th>Task</th>
<th>Rate</th>
<th>Service time</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1/40ms</td>
<td>5ms</td>
<td>30ms</td>
</tr>
<tr>
<td>B</td>
<td>1/100ms</td>
<td>?ms</td>
<td>100ms</td>
</tr>
<tr>
<td>C</td>
<td>1/50ms</td>
<td>10ms</td>
<td>25ms</td>
</tr>
</tbody>
</table>

Please answer the following two questions assuming the system has a weak priority system.

(A) (1 point) What is the maximum service time for task B that still allows all constraints to be met?

Task C has max latency of 15 ms ⇒ limits service time of task B.

Maximum service time for task B (ms): 15 if \( C > A > B \) in (8)

10 if \( A > C > B \) in (6)

(B) (1 point) Assuming a suitable service time for B, give a weak priority order for the tasks that meets the constraints. List the task with the highest priority first, the task with the lowest priority last.

earliest deadline first!

Weak priority order for the tasks: \( C > A > B \)

Please answer the following two questions assuming the system has a strong priority system where task C has the highest priority and task B has the lowest priority.

(C) (2 points) What is the maximum service time for task B that still allows all constraints to be met?

In 100 ms worst-case:

3 task A interrupts (15 ms total)

2 task C interrupts (20 ms total)

35 ms service time max ⇒ 100 - 35 = 65 ms available for B

Maximum service time for task B (ms): 65

(D) (3 points) Assume B’s service time is 10ms. For each task, what is the worst-case delay between the task’s interrupt and completion of the task’s interrupt handler?

priority: \( C > A > B \)

Worst-case delay for task A (ms): 15

wait for \( C \) service time

start immediately = \( C \) service time

Worst-case delay for task B (ms): 25

Worst-case delay for task C (ms): 10

wait for \( A \) service time

(10+5) + 10 = 25
Problem 4. Semaphores (9 points)

The following pair of processes share the variable counter, which has been given an initial value of 10 before execution of either process begins:

<table>
<thead>
<tr>
<th>Process A</th>
<th>Process B</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>A1: LD(counter,R0)</td>
<td>B1: LD(counter,R0)</td>
</tr>
<tr>
<td>ADDC(R0,1,R0)</td>
<td>ADDC(R0,2,R0)</td>
</tr>
<tr>
<td>A2: ST(R0,counter)</td>
<td>B2: ST(R0,counter)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

(A) (3 points) If Processes A and B are run on a timesharing system, there are six possible orders in which the LD and ST instructions might be executed. For each of the orderings, please give the final value of the counter variable.

A1 A2 B1 B2: counter = 13
A1 B1 A2 B2: counter = 12
A1 B1 B2 A2: counter = 11
B1 A1 B2 A2: counter = 11
B1 A1 A2 B2: counter = 12
B1 B2 A1 A2: counter = 13

In the following two questions you are asked to modify the original programs for processes A and B by adding the minimum number of semaphores and signal and wait operations to guarantee that the final result of executing the two processes will be a specific value for counter. Give the initial values for every semaphore you introduce. For full credit, your solution should allow all execution orders that result in the required value.

(B) (3 points) Add semaphores (with initial values) so that the final value of counter is 12.

Semaphores: \( X = 0, Y = 0 \)

Process A      Process B      
...            ...            
A1: LD(counter,R0)  B1: LD(counter,R0)  
ADD(R0,1,R0) \( \text{WAIT}(X) \)  ADDC(R0,2,R0) \( \text{SIGNAL}(X); \text{WAIT}(Y) \)  
A2: ST(R0,counter)  B2: ST(R0,counter)  
\( \text{SIGNAL}(Y) \)  ...

(C) (3 points) Add semaphores (with initial values), so that the final value of counter is not 13.

Semaphores: \( X = 0, Y = 0 \)

Process A          Process B          
...              ...              
A1: LD(counter,R0)  B1: LD(counter,R0)  
\( \text{SIGNAL}(X) \)  ADDC(R0,2,R0) \( \text{WAIT}(X) \)  
A2: ST(R0,counter)  B2: ST(R0,counter)  
\( \text{WAIT}(Y) \)  ...

END OF QUIZ 4! Have a good summer...