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.

Problem 1. (10 Points) Virtual Memory

(4 Points) Consider a virtual memory system that uses a single-level page map to translate virtual addresses into physical addresses. Each of the questions below asks you to consider what happens when just ONE of the design parameters (page size, virtual memory size, physical memory size) of the original system is changed. Circle the correct answer.

(A) If the physical memory size (in bytes) is doubled, the number of entries in the page table 
   (a) stays the same
   (b) doubles
   (c) is reduced by half
   (d) increases by one
   (e) decreases by one
   
(B) If the page size (in bytes) is halved, the number of entries in the page table 
   (a) stays the same
   (b) doubles
   (c) is reduced by half
   (d) increases by one
   (e) decreases by one
   
(C) If the virtual memory size (in bytes) is doubled, the number of bits in each entry of the page table 
   (a) stays the same
   (b) doubles
   (c) is reduced by half
   (d) increases by one
   (e) decreases by one
   
(D) If the page size (in bytes) is doubled, the number of bits in each entry of the page table 
   (a) stays the same
   (b) doubles
   (c) is reduced by half
   (d) increases by one
   (e) decreases by one
   
   VPN size unchanged
   PPN size is one bit smaller
Consider a virtual memory system for the Gamma processor with 4096 \((2^{12})\) virtual pages and 16384 \((2^{14})\) physical pages where each page contains 1024 \((2^{10})\) bytes. The first 8 entries of the current page map are shown below:

<table>
<thead>
<tr>
<th>Index</th>
<th>D</th>
<th>R</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0x22</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0x01</td>
</tr>
<tr>
<td>2</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0x03</td>
</tr>
<tr>
<td>5</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0x15</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

(E) (1 Point) What is the total number of bits in the page map?

\[
\text{Total number of bits in the page map: } 2^{14} \cdot 2^{12} = 2^{26}
\]

(F) (2 Points) Which address bits from the CPU are used to choose an entry from the page table?

Address bits used to choose page table entry: A[21 : 10]

(G) (2 Points) What is the physical address for the word at virtual location 0x1234? Write “not resident” if the location is not currently present in physical memory.

Physical address for byte at virtual address 0x1234 or “not resident”: 0x E34

(H) (1 Point) Briefly explain what action caused the D bit for page 6 to be 1.

Briefly explain.

A ST instruction wrote to a location in virtual page 6.
**Problem 2.** (5 Points) SVCs

In lecture we arrived at the following implementation for the ReadKey supervisor call, which waits until there is a character available in the keyboard buffer, then returns it to the user in R0. Three lines in the handler have been labeled [A], [B], and [C].

```c
ReadKey_h() {
    int kbdnum = ProcTbl[Cur].DPYNum;
    if (BufferEmpty(kbdnum)) {
        [B] Scheduler();
    } else {
        [C] User.Reg[0] = ReadInputBuffer(kbdnum);
    }
}
```

Below we'll consider the effect of removing each labeled line in turn. Please choose one of the following as the best characterization of the effect of removing the line:

1. Execution appears to halt as there is now a loop in Kernel mode.
2. Both the requesting process and other processes run as before.
3. The requesting process runs as before; other processes receive a smaller percentage of the CPU time.
4. The requesting process runs as before; other processes receive a larger percentage of the CPU time.
5. The requesting process receives an incorrect character.

(A) (1 Point) Line [A] is removed from the handler.

**Effect on execution? (circle one) 1 ... 2 ... 3 ... 4 ... 5**

(B) (1 Point) Line [B] is removed from the handler.

**Effect on execution? (circle one) 1 ... 2 ... 3 ... 4 ... 5**

(C) (1 Point) Line [C] is removed from the handler.

**Effect on execution? (circle one) 1 ... 2 ... 3 ... 4 ... 5**

(D) (2 Points) A summer intern decides that if a process is waiting for a character it should be scheduled for execution half as often, so he adds a second call to `Scheduler()`, i.e., he duplicates line [B]. Briefly describe the actual effect of this change. To be concrete assume there are N processes and that it’s process 0 that executes the ReadKey SVC.

*While process 0 is waiting for a character, process 1 is never scheduled.*

Brief description

6.004 Fall 2015 - 3 of 6 - Quiz #4
Problem 3. (6 Points) Interrupts & Realtime

A computer system has three devices whose characteristics are summarized in the following table:

<table>
<thead>
<tr>
<th>Device</th>
<th>Service time</th>
<th>Interrupt Frequency</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>400us</td>
<td>1/(800us)</td>
<td>800us</td>
</tr>
<tr>
<td>D2</td>
<td>250us</td>
<td>1/(1000us)</td>
<td>300us</td>
</tr>
<tr>
<td>D3</td>
<td>100us</td>
<td>1/(800us)</td>
<td>400us</td>
</tr>
</tbody>
</table>

Service time indicates how long it takes to run the interrupt handler for each device. The maximum time allowed to elapse between an interrupt request and the end of the execution of the interrupt handler is indicated by the deadline.

A. (2 points) If a user-mode program P takes 100 seconds to execute when interrupts are disabled, approximately how long will P take to run when interrupts are enabled?

Approximate time for P to run with interrupts enabled (seconds): $\frac{8 \times 100}{800} = 800$

$D1: \frac{400}{800} = 50\%$ of cpu time

$D2: \frac{250}{1000} = 25\%$

$D3: \frac{100}{800} = 12.5\%$

\[
\text{only } 12.5\% = \frac{1}{8} \text{ of cpu time for running P.}
\]

B. (2 points) Can the requirements given in the table above be met using a weak priority ordering among the interrupt requests? If so give priority ordering for D1, D2, D3 or list device(s) whose deadlines cannot be met.

Weak priority ordering or list device(s) with missed deadlines: D2, D3

if D1 is running, D2 & D3 may miss deadlines since D2 max latency is 50us and D3 max latency is 300us.

C. (2 points) Can the requirements given in the table above be met using a strong priority ordering among the interrupt requests? If so give priority ordering for D1, D2, D3 or list device(s) whose deadlines cannot be met.

Strong priority ordering or list device(s) with missed deadlines: D2 > D3 > D1
**Problem 4.** (9 Points) Semaphores

P1 and P2 are processes that run concurrently. P1 has two sections of code where section A is followed by section B. Similarly, P2 has two sections: C followed by D. Within each process execution proceeds sequentially, so we are guaranteed that $A \leq B$, i.e., A precedes B. Similarly we know that $C \leq D$. There is no looping; each process runs exactly once. You will be asked to add semaphores to the programs – you may need to use more than one semaphore. Please give the initial values of any semaphores you use. For full credit use a minimum number of semaphores and don’t introduce any unnecessary precedence constraints.

(A) (3 points) Please add WAIT(...), SIGNAL(...) statements as needed in the spaces below so that the precedence constraint $B \leq C$ is satisfied, i.e., execution of P1 finishes before execution of P2 begins.

**Add WAIT and SIGNAL statements so that $B \leq C$**

Semaphore initial values: $S = 0$

<table>
<thead>
<tr>
<th>Process P1</th>
<th>Process P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>...Section A code...</td>
<td>wait($S$)</td>
</tr>
<tr>
<td>...Section B code...</td>
<td>... Section C code...</td>
</tr>
<tr>
<td>signal($S$)</td>
<td>...Section D code...</td>
</tr>
</tbody>
</table>

(B) (3 points) Please add WAIT(...), SIGNAL(...) statements as needed in the spaces below so that $D \leq A$ or $B \leq C$, i.e., executions of P1 and P2 cannot overlap, but are allowed to occur in either order.

**Add WAIT and SIGNAL statements so that $D \leq A$ or $B \leq C$**

Semaphore initial values: $M = 1$

<table>
<thead>
<tr>
<th>Process P1</th>
<th>Process P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>wait($M$)</td>
<td>wait($M$)</td>
</tr>
<tr>
<td>...Section A code...</td>
<td>... Section C code...</td>
</tr>
<tr>
<td>...Section B code...</td>
<td>...Section D code...</td>
</tr>
<tr>
<td>signal($M$)</td>
<td>signal($M$)</td>
</tr>
</tbody>
</table>
(C) (3 points) Please add WAIT(...) and SIGNAL(...) statements as needed in the boxes below so that A ≤ D and C ≤ B, i.e., the first section (A and C) of both processes completes execution before the second section (B or D) of either process begins execution.

**Add WAIT and SIGNAL statements so that A ≤ D and C ≤ B**

Semaphore initial values: $S=0$, $T=0$

<table>
<thead>
<tr>
<th>Process P1</th>
<th>Process P2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Section A</strong></td>
<td><strong>Section C</strong></td>
</tr>
<tr>
<td><code>signal(S)</code></td>
<td><code>signal(T)</code></td>
</tr>
<tr>
<td><code>wait(T)</code></td>
<td><code>wait(S)</code></td>
</tr>
</tbody>
</table>

**Section B**

**Section D**

END OF QUIZ 4 – have a good break!

WERF!