Prelim Interlude
32bit architecture, byte addressable. The stack grows down from high to low addresses

Addresses from 0x0 to 0xFFFF FFFF
32-bit architecture, byte addressable. The stack grows down from high to low addresses.

What is the address of the next to last word?

Addresses from 0x0 to 0xFFFF FFFF
32-bit architecture, byte addressable. The stack grows down from high to low addresses.

What is the address of the next to last word?

Addresses from 0x0 to 0xFFFF FFFF
32bit architecture, byte addressable. The stack grows down from high to low addresses.

What is the address of the next to last word?

Addresses from 0x0 to 0xFFFF FFFF

0xFFFF FFFF8
If the stack pointer is at 0xFFFF 8D70 and three words are pushed onto the stack, what is the new value?

32bit architecture, byte addressable. The stack grows down from high to low addresses.

3 words = 12 bytes downward

0xFFFFE 8D70 - 0xB = 0xFFFFE 8D64
32-bit architecture, byte addressable. The stack grows down from high to low addresses.

User process occupies bottom half of the 32-bit address space (i.e., the lower addresses), while the kernel occupies the top half of the same address space (i.e., the higher addresses).

What is the address where the kernel starts (i.e. the lowest address in the kernel)?

0xFFFF FFFF / 2

Right shift by one position
32bit architecture, byte addressable. The stack grows down from high to low addresses.

User process occupies bottom half of the 32-bit address space (i.e., the lower addresses), while the kernel occupies the top half of the same address space (i.e., the higher addresses).

What is the address where the kernel starts (i.e. the lowest address in the kernel)?

0xFFFF FFFF / 2

Right shift by one position
32-bit architecture, byte addressable. The stack grows down from high to low addresses.

User process occupies bottom half of the 32-bit address space (i.e., the lower addresses), while the kernel occupies the top half of the same address space (i.e., the higher addresses).

What is the address where the kernel starts (i.e. the lowest address in the kernel)?

$$0x7FFF FFFF + 1 = 0x8000 0000$$
32-bit architecture, byte addressable. The stack grows down from high to low addresses.

User process occupies bottom half of the 32-bit address space (i.e., the lower addresses), while the kernel occupies the top half of the same address space (i.e., the higher addresses).

What is the address of the last byte of a user process (i.e. the highest user space address)?
What is the value of the user stack pointer?

0x7FF277E0
The table on the right shows the interrupt (trap) vector. Suppose the CPU is executing in user space. The program counter is 0x000006FC8, the user stack pointer is 0x7FF277E0, and the kernel stack pointer is 0xFFFFA37C0. Now a disk interrupt occurs, pushing the user’s PC, SP, and PSW onto the kernel stack. The interrupt handler pushes 5 general purpose registers onto the stack. Each push instruction occupies 4 bytes.

Answer the following questions about the CPU’s state at this moment:

What is the value of the kernel stack pointer?

KSP starts at 0xFFFFA37C0

Push \{ User’s PC, SP, PSW
       5 general purpose registers

8 words — downwards

0xFFFFA37C0 - 0x20 = 0xFFFFA37A0
What is the value of value of the PC?

PC starts at 0x80001079

5 push instructions, each 4 bytes = 20 bytes

0x80001079 + 0x14 = 0x8000108D
The table on the right shows the interrupt (trap) vector. The program counter is 0x00006FC8, the user stack pointer is **0x7FF277E0**, and the kernel stack pointer is **0xFFF37C0**. Now a disk interrupt occurs, pushing the user’s PC, SP, and PSW onto the kernel stack. The interrupt handler pushes 5 general purpose registers onto the stack. Each push instruction occupies 4 bytes.

Answer the following questions about the CPU’s state at this moment:

<table>
<thead>
<tr>
<th>Index</th>
<th>Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x806ECEB0</td>
<td>System Call</td>
</tr>
<tr>
<td>1</td>
<td>0x8050D330</td>
<td>Divide-by-Zero</td>
</tr>
<tr>
<td>2</td>
<td>0x8024B280</td>
<td>Page Fault</td>
</tr>
<tr>
<td>3</td>
<td>0x8001079</td>
<td>Disk Interrupt</td>
</tr>
<tr>
<td>4</td>
<td>0x8052C420</td>
<td>Clock Interrupt</td>
</tr>
</tbody>
</table>

Now, suppose the disk interrupt handler has completed and has just executed the return-from-interrupt instruction. Answer the following questions about the CPU state at that time.

What is the value of the user SP?

**0x7FF277E0**
The table on the right shows the interrupt (trap) vector.

Suppose the CPU is executing in user space. The program counter is 0x00006FC8, the user stack pointer is 0x7FF277E0, and the kernel stack pointer is 0xFFFFA37C0. Now a disk interrupt occurs, pushing the user’s PC, SP, and PSW onto the kernel stack. The interrupt handler pushes 5 general purpose registers onto the stack. Each push instruction occupies 4 bytes.

Answer the following questions about the CPU’s state at this moment:

Now, suppose the disk interrupt handler has completed and has just executed the return-from-interrupt instruction. Answer the following questions about the CPU state at that time.

What is the value of the kernel SP?

0xFFFFA37C0
The table on the right shows the interrupt (trap) vector. Suppose the CPU is executing in user space. The program counter is 0x00006FC8, the user stack pointer is 0x7FF27EE0, and the kernel stack pointer is 0xFFF937C0. Now a disk interrupt occurs, pushing the user’s PC, SP, and PSW onto the kernel stack. The interrupt handler pushes 5 general purpose registers onto the stack. Each push instruction occupies 4 bytes.

Answer the following questions about the CPU’s state at this moment:

Now, suppose the disk interrupt handler has completed and has just executed the return-from-interrupt instruction. Answer the following questions about the CPU state at that time.

What is the value of the PC?

0x00006FC8
How many times will the value of result be printed?

First value(s)? Last value(s)?
How many times will the value of result be printed?

First value(s)? Last value(s)?
How many times will the value of result be printed?

First value(s)? Last value(s)?
How many times will the value of result be printed?

First value(s)? Last value(s)?