Interrupts without Context switching

Interrupts without context switching
Interrupts without Context switching

Assuming that process $P_1$ is executing and a keyboard interrupt occurs:

1. CPU (hardware) pushes the current PC onto the stack and updates the PC-register to contain the address of the 1$^{st}$ instruction in the corresponding interrupt handler.
2. The interrupt handler starts execution by pushing all other registers onto the stack.
3. Performs its specific task that modifies the CPU registers content.
4. Pops back the registers values from the stack into the CPU.
5. The handler executes RTI (ReTurn from Interrupt) which cause the CPU (hardware) to pop PC values from stack to CPU’s PC register.
6. The CPU then, resumes executing process $P_1$. 
Example of Interrupt without Context switching (Con’t)
Example of Interrupt without Context switching (Con’t)

```
0x100 Mov Ax, 13
0x101 Mov Bx, 7
0x102 Add Ax, Bx

0xB100 Mov Ax, 10
0xB101 Mov Bx, 20
0xB102 Mul Ax, Bx

0x300 Does...
0x3 Echo char
0x3 Does...
0x3FF RTI

0x400 Does...
0x4 Context Switching
0x4 Does...
0x600 RTI
```

Karim Lounis (Queen's University, Canada)
Example of Interrupt without Context switching (Con’t)
Example of Interrupt without Context switching (Con’t)

0xB100 Mov Ax, 13
0x9100 Mov Bx, 7
0x9102 Add Ax, Bx

0xB100 Mov Ax, 10
0xB101 Mov Bx, 20
0xB102 Mul Ax, Bx

0x0300 Does...
0x03 Echo char
0x03 Does...
0x03FF RTI

0x0400 Does...
0x04 Context Switching
0x04 Does...
0x0600 RTI

0x01C02
0xEBBB2
0xEB100

Hard Disc Drive

PID: Status
PC: 0xB100
Registers:
Resources:
Priority

PID: Status
PC: 0x9102
Registers:
Resources:
Priority

Operating System Kernel

Other interrupt handler
0x03FF
0x0300
Keyboard interrupt handler
0x0600
0x0400
Timer interrupt handler

PCBs

0x0000

0x0001

0x0002

RAM

0xFFFF

0xFFFD

0xFFF

0x1000

0x0000

0x0B00

0x0C00
Example of Interrupt without Context switching (Con’t)

The user types one character in the keyboard.
Example of Interrupt without Context switching (Con’t)

The CPU detects the interrupt and:
(1) Pops the PC onto the process stack

Karim Lounis (Queen’s University, Canada)
Example of Interrupt without Context switching (Con’t)

The CPU detects the interrupt and:
(1) Pops the PC onto the process stack
Example of Interrupt without Context switching (Con’t)

The CPU detects the interrupt and:
1. Pops the PC onto the process stack
2. Loads the PC with the address of the handler's first instruction
Example of Interrupt without Context switching (Con’t)

The CPU detects the interrupt and:
(1) Pops the PC onto the process stack
(2) Loads the PC with the address of the handler's first instruction

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1000</td>
<td>MOV AX, 13</td>
</tr>
<tr>
<td>0x1010</td>
<td>MOV BX, 7</td>
</tr>
<tr>
<td>0x1020</td>
<td>ADD AX, BX</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0xB100</td>
<td>MOV AX, 10</td>
</tr>
<tr>
<td>0xB101</td>
<td>MOV BX, 20</td>
</tr>
<tr>
<td>0xB102</td>
<td>MUL AX, BX</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0300</td>
<td>... Echo char</td>
</tr>
<tr>
<td>0x0303</td>
<td>... Does</td>
</tr>
<tr>
<td>0x03FF</td>
<td>RTI</td>
</tr>
<tr>
<td>0x0400</td>
<td>... Does</td>
</tr>
<tr>
<td>0x0404</td>
<td>Context Switching</td>
</tr>
<tr>
<td>0x0404</td>
<td>... Does</td>
</tr>
<tr>
<td>0x0600</td>
<td>RTI</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>0x0FEEF</td>
</tr>
<tr>
<td>Stack Top</td>
<td>0x0000</td>
</tr>
<tr>
<td>Stack Bottom</td>
<td>0x0FF00</td>
</tr>
</tbody>
</table>

Karim Lounis (Queen’s University, Canada)
Example of Interrupt without Context switching (Con’t)

Context Switching

Karim Lounis (Queen’s University, Canada)
Example of Interrupt without Context switching (Con’t)
Example of Interrupt without Context switching (Con’t)

Karim Lounis (Queen’s University, Canada)
Example of Interrupt without Context switching (Con’t)

- **PID**: Status
- **PC**: 0xB100
- **Registers**: Resources Priority

- **PID**: Status
- **PC**: 0x0102
- **Registers**: Resources Priority

- **Stack**
  - 0xB102
  - AX
  - BX
  - ...
  - BP
  - FLAGS

- **Hard Disc Drive**
- **IP**: 0x0300
- **IR**: save state
- **Pushes all registers onto the interrupted process stack**

- **Example Code**:
  - 0x9100 Mov Ax, 13
  - 0x9101 Mov Bx, 7
  - 0x9102 Add Ax, Bx
  - ...
  - 0xB100 Mov Ax, 10
  - 0xB101 Mov Bx, 20
  - 0xB102 Mul Ax, Bx
  - ...

- **Operating System Kernel**
  - 0x03FF RTI
  - 0x0400 Does ...
  - 0x04 Context Switching
  - 0x04 Does ...
  - 0x0600 RTI
Example of Interrupt without Context switching (Con’t)

0x9100 Mov Ax, 13
0x9101 Mov Bx, 7
0x9102 Add Ax, Bx
......
0xB100 Mov Ax, 10
0xB101 Mov Bx, 20
0xB102 Mul Ax, Bx
......
0x0300 Does...
0x03 . Echo char
0x03 . Does...
0x03FF RTI
0x0400 Does...
0x04 . Context Switching
0x04 . Does...
0x0600 RTI

Other interrupt handler
0x03FF
0x0300
Keyboard interrupt handler
0x0600
0x0400
Timer interrupt handler

PID: Status
PC: 0xB100
Registers: Resources
Priority

PID: Status
PC: 0x9102
Registers: Resources
Priority

Stack

0xB102
AX
BX
BP
FLAGS

Hard Disc Drive

IP: 0x03xx
IR: Does...
Example of Interrupt without Context switching (Con’t)

The handler restores the registers (pops) from the stack into the CPU.

0x9100 Mov Ax, 13
0x9101 Mov Bx, 7
0x9102 Add Ax, Bx
...
...
0xB100 Mov Ax, 10
0xB101 Mov Bx, 20
0xB102 Mul Ax, Bx
...
...
0x0300 Does...
0x03 Echo char
0x03 Does...
0x03FF RTI
0x0400 Does...
0x04 Context Switching
0x04 Does...
0x0600 RTI
Example of Interrupt without Context switching (Con’t)

Hard Disc Drive

Mozilla Firefox.exe
Skype.exe
Notepad++.exe

PID: Status
PC: 0xB100
Registers Resources Priority

PID: Status
PC: 0xB102
Registers Resources Priority

0x9100 Mov Ax, 13
0x9101 Mov Bx, 7
0x9102 Add Ax, Bx
...

0xB100 Mov Ax, 10
0xB101 Mov Bx, 20
0xB102 Mul Ax, Bx
...

0x0300 Does...
0x03 Echo char
0x03 Does...
0x03FF RTI

0x0400 Does...
0x04 Context Switching
0x04 Does...
0x0600 RTI

0xB102
AX
BX
...
BP
FLAGS
Stack

Operating System Kernel

IP: 0x03FF
IR: RTI
Example of Interrupt without Context switching (Con’t)

- **0xB102**
  - AX
  - BX
  - ...
  - BP
  - FLAGS

Stack

- **RAM**
  - 0xFFFF
  - 0xFFF
  - 0xB000
  - 0xC000
  - 0x0900

- **Operating System Kernel**
  - 0x03FF RTI
  - 0x0600 RTI
  - 0x0400 Does ...
  - 0x04 Does ...
  - 0x03 Does ...
  - 0x03 Does ...
  - 0x03 Does ...

- **PID**: Status
- **PC**: 0xB100
- **Registers**: 0
- **Resources**: Priority

- **Hard Disc Drive**

- **The CPU pops the PC value from the stack**

- **Mozilla Firefox.exe**
- **Skype.exe**
- **Notepad++.exe**

- **PC**: 0x9100
- **Mov Ax, 13**
- **Mov Bx, 7**
- **Add Ax, Bx**

- **PC**: 0x9102
- **Mov Ax, 10**
- **Mov Bx, 20**
- **Mul Ax, Bx**
Example of Interrupt without Context switching (Con’t)

- Mozilla Firefox.exe
- Skype.exe
- Notepad++.exe

PID: Status PC: 0xB102
Registers
Resources
Priority

PID: Status PC: 0xB102
Registers
Resources
Priority

The CPU pops the PC value from the stack

- Hard Disc Drive

- 0xB102
- AX
- BX
- ...
- BP
- FLAGS

Stack

- 0x0000
- 0x0600
- 0x03FF
- Other interrupt handler
- 0x0300
- Keyboard interrupt handler
- 0x0400
- Timer interrupt handler

0x9100 Mov Ax, 13
0x9101 Mov Bx, 7
0x9102 Add Ax, Bx

0xB100 Mov Ax, 10
0xB101 Mov Bx, 20
0xB102 Mul Ax, Bx

0x0300 Does...
0x03 Echo char
0x03 Does...
0x03FF RTI

0x0400 Does...
0x04 Context Switching
0x04 Does...
0x0600 RTI

Karim Lounis (Queen’s University, Canada)
Example of Interrupt without Context switching (Con’t)

PID: Status
PC: 0xB100
Registers: Resources
Priority

Stack

Hard Disc Drive

Mozilla Firefox.exe
Skype.exe
Notepad++.exe

0xB102
AX
BX
...
BP
FLAGS

0xB102 Mov Ax, 13
0xB101 Mov Bx, 7
0xB102 Add Ax, Bx
...

0xB100 Mov Ax, 10
0xB101 Mov Bx, 20
0xB102 Mul Ax, Bx
...

0x0300 Does...
0x03 Echo char
0x03 Does...
0x03FF RTI

0x0400 Does...
0x04 Context Switching
0x04 Does...
0x0600 RTI

IP: 0xB102
IR: Mul Ax, Bx

Karim Lounis (Queen’s University, Canada)
The occurrence of an interrupt does not always imply a context switch:

1. Keyboard interrupts.
2. DMA completion interrupts.
3. Raster interrupts.

Other interrupts cause a context switch:

1. System calls.
2. Exceptions.
3. Timer interrupt.

Of course, if there is only one process, then there is no context switching even with timer interrupt.
Interrupt handler and Context Switching

Generally, right after an interrupt (hardware or software):

1. The CPU pushes the PC value into the stack pointed by SP.
2. The CPU loads the PC register with a new value (@ of the handler).
3. The handler saves the remaining registers onto the stack.
4. The handler executes its specific task.
5. The handler determines whether a context switching is needed or not.

If yes, a context switching is performed. The previously pushed registers are saved into the PCB (of the interrupted process) and the CPU is loaded with a new content, except the PC, from a new PCB.

If no, goto 6.

6. The handler Executes RTI.
7. The CPU pops back the PC value from the stack pointed by SP.
Interrupts Interrupt Interrupt handlers

During the execution of an interrupt handler, other interrupts can occur. Interrupts are executed in a nested order i.e., if an interrupt of a higher priority occurs while another interrupt handler of a lower priority is executing, the current interrupt handler is interrupted.

When an interrupt handler is interrupted, its PC value is pushed by the CPU onto the stack dedicated for interrupts (located in kernel space). The new handler saves the other register into that interrupts stack. The interrupts stack is pointed by the ISP (Interrupts Stack Pointer). Once the new handler terminates, it pops the registers of the old handler. Executes RTI, which informs the CPU to pop the PC value from interrupts stack so that the old handler can resume execution.
End