Context Switch in OS

What is Context Switch ?

What are steps involved in context switch ?

When does context switch happen ?

Performance of Context switch ?

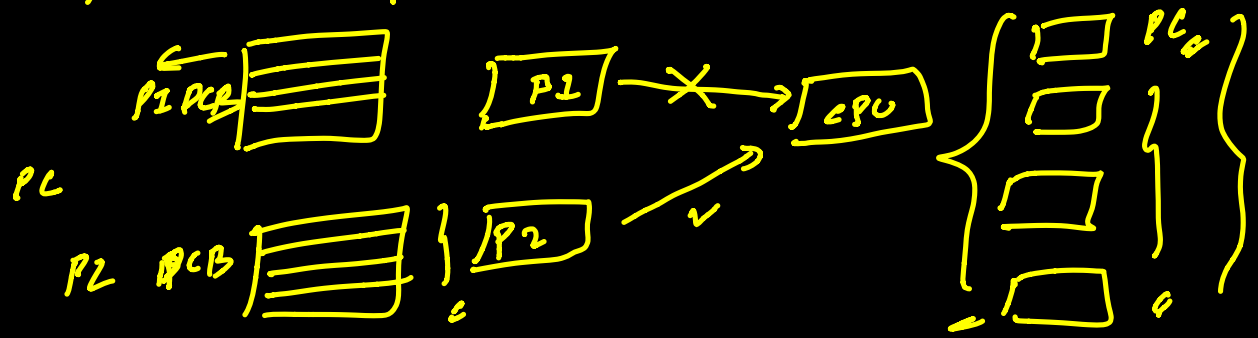
What is context switch ?

When the CPU needs to switch to a different process than the one that is currently running, the kernel needs to save the state of current process and restore the state of the scheduled process. This task of saving and restoring state is called context switch.

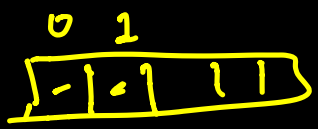
Steps involved in Context switch

Processor state change

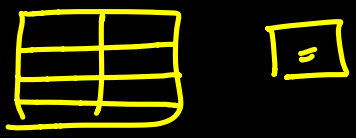
- Save the values of the CPU registers that the Current program was using in its PCB.
- Restore the register content from the PCB of scheduled process to the CPU registers.



Virtual memory mapping switch



- This involves the switch of a register value in CPU where the base address of the page table is stored.



- Unless the Context switch is happening between two threads of the same process, this step is involved.
- Threads with in a process share the virtual memory and hence sharing the page tables.

When does the Context switch happen?

In multitasking environment

- Each process only gets a fixed time to run on a processor. Once this time expires, the process Scheduler removes this process and schedules other.
- If a process is waiting for an I/O event or on any other event, the process scheduler marks its state as waiting and schedules another one

Interrupt handling (Hardware interrupts)

- when an interrupt occurs, the processor interrupts the process running on it and calls an interrupt handler to handle the interrupt.
- Execution mode switches from user to Kernel mode Since interrupt handlers are a part of Kernel code.
- Since the interrupt is not related to the process that is running on processor, the interrupt handler saves the state of current process and executes its code on processor.
- when exiting from the interrupt handler, the state is restored and the control is returned to the interrupted process.

Switching between user and kernel mode

- When a user program needs to use a system call, the switch is needed.
- This switch is achieved by using a software interrupt called trap.
- When using trap, a particular instruction called INT is executed on the processor which switches to kernel mode.
- The system call runs in the process context itself. Upon exiting the kernel, the process resumes in user mode.
- The time taken for the switch from user to kernel mode is very less as it only involves very few instructions.
- Also, this is not a Context switch.

- But sometimes, a system call may result to Context switch (read/write, wait()).

Performance cost of Context Switching

- Context switch time are highly dependent on hardware support.
- If the CPU has multiple set of registers or if the Kernel uses single page table for all processes, context switch time will be very less.

The overhead of copying register values

- Copying CPU register values to PCB and PCB's register values back to CPU registers is an overhead.
- This may take some milliseconds.

TLB (Translation lookaside buffer) miss

- Translating a Virtual address to physical address is expensive since it requires multiple RAM access (page tables are stored in RAM)

- Most modern processor uses a cache to store the mappings that were already accessed.
- This cache is called TLB.
- when the page table changes, the TLB or the cache gets flushed.
- This will lead to increase in access of page tables from RAM.

1. Context switch time due to multitasking ~ t_1
2. Context switch time during interrupt handling ~ t_2
3. Time to switch from user to kernel mode. ~ t_3

$1 \geq 2 > 3$

$t_1 \geq t_2 > t_3$