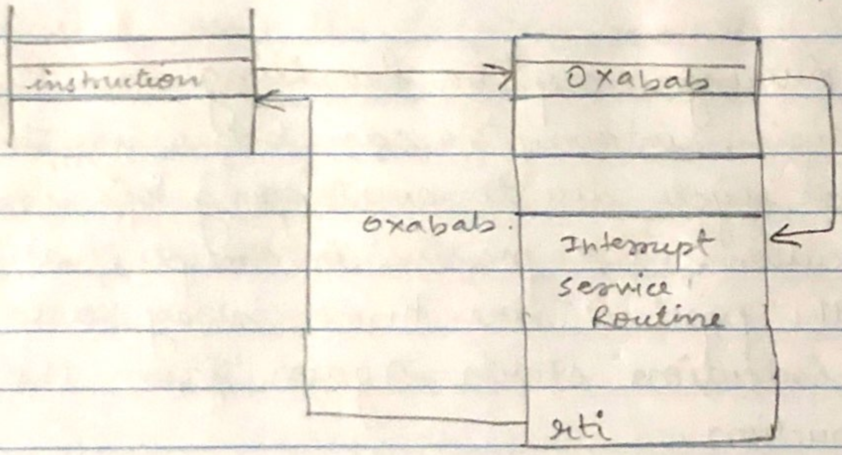
TANU SHREE

OS HOMEWORK - WEEK-2

1. a) The instruction 'rti' is used at the end of the interrupt service routine. Interrupt Service routines do not return using a return-from-subroutine instruction, but rather this 'routine-from interrupt' also called 'rti'. It is used at the end of the contextswitch where the state of CPU is automatically restored and the program counter is loaded to continue the execution where the interrupt happened.



b.) The instruction 'rti' is a privileged instruction and it is usually only available while the machine is running in supervisor mode. Application program runs in user mode and if it executes 'rti' instruction, it is trapped by the Hardware in the operating system, ie it is hardware exception. The instruction will not be executed and will be treated as an illegal instruction.

2. a.) Large number of registers increases the size of instruction and the area used by the core. Saving and restoring the registers across the function calls, system calls and interrupts take extra time and memory bandwidth. This results in making context switch expensive. Also, large number of registers are expensive hardware and most of them can remain underutilized.

b.) During exception handling in Pipelined Processors, multiple interrupts can occur in the same clock cycle due to overlapping of instruction execution. For precise interrupts, all the instruction in the pipeline are thrown away and then the execution starts again from the faulting instruction.

Deeper pipeline will take longer to flush out the instructions, resulting in the increase of latency in executing the interrupts.

Longer pipeline introduces more dependencies. Pipeline registers between each stage have sequencing overhead due to which there are diminishing returns.

Deep pipelines can in result increase the cost of context switching overhead.

3. Yes, there would be a need of critical section within that kernel. When user invokes the system call, the system security is naturally enforced, as the system call handler will check whether the calling user program is authorized to make the requested call or not.

While the operating system is running, the machine receives an interrupt when the user invokes a system call. In order to protect the ongoing execution by kernel from the interrupt handler, the interrupt needs to be masked. Now, the thread is executing in the kernel which is modifying some global data structure. If the mutual exclusion by masking the interrupts is not enforced in kernel, system call interrupt handler may modify the data structure. Hence, critical section is needed in the kernel.