Prashant Singh Chandel
2016PH10566.

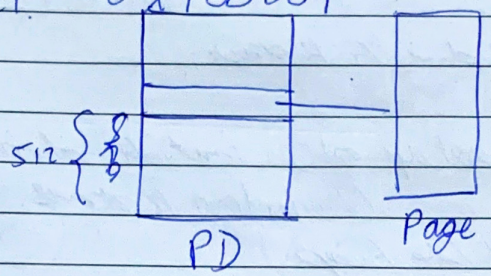
Homework-3

Assignment: Paging

The page directory entry = VA >> 22 = 512 = 0x200
offset

The page i in the 257th page in that page directory (offset of page) = 0x100

The writable bit must be set to 0. and present bit should be! 0x100001



Assignment: Page table reload.

kpgdir[0] is 0x0 because we have already executed setupkvm() and it has mapped only the entries above kernbase.

VA = 0x80107beb

VA → PA would be just VA - kernbase i.e 0x80000000
= 0x107beb

0x80107beb >> 22.

$4 0x200 → this is the Page directory offset.

PDE → 0x114007 here 7 corresponds to 111 last three digits which means the present (last bit), writable (second last) and user bits are set.

0x107001 here 1 means the page is present
0x107000 corresponds to lowest address in the page.

physical address works in ~~gdb~~ gdb because switch_kvm() hasn't executed yet and the physical address has 2 mappings one above kernbase and one VA=PA.

After switch_kvm() it switches to kernel space and the earlier mapping is removed so 0x107 ~~doesn't~~ doesn't work now.

Assignment: Addressing

The ELF specifies the address when it needs to be loaded and by just changing the address in bootmain by adding 0x100000 it would create problems because the code that will run would expect it the kernel to be loaded from 1MB ~~and changed~~ not from 2MB This will cause problems in booting.

Traps → Three saved registers can be present in case of 1 system call, one external interrupt and then it context switches on the second interrupt.

i) → No, 2 context structures can't be present as interrupts are disabled while context switching ~~and~~ and the interrupts are enabled only after the context switch has taken place.

ii) Yes, Two trapframes can be present as 1 trapframe and one context. 2 trapframes and one context can also be present. as interrupts would be disabled on This is the case where 3 registers will be present.

This happens when process context switches while executing second interrupt the second tf ex timer interrupt can occur. If the external interrupt handler

ii.) No, more than 3 sets of saved registers can't be present as the interrupts are disabled both for external interrupt handler and context switch. This is done as a precaution because the stack depth is limited.

Assignment: Context Switching

sched() executes on the stack of any process thread. It is allocated from the Kernel's heap.

Scheduler() runs on a seperate stack that is not mapped to any process and doesn't have a user space. Each processes has a scheduler stack which is also allocated from kernel's heap.

when sched() calls swtch() that swtch() doesn't immediately returns. The scheduler assigns another process to run. After that when that process calls swtch() and scheduler decides to continue the execution on this process then it returns. If there is no other process then it returns immediately.

The pattern is. cbad which occurs again and again.

The first 2 characters are ac because the scheduler runs first which then switches to a process [2] which eventually calls sched() [1] and returns to the scheduler.

We can make switch more efficient by using a special instruction that ~~loads~~ saves all the old registers and loads the new registers.