AKHILESH KARRA 2015CS10233

COL331 OS HW-3

Q1

Paging

VA ⇒ 1000000000 0100000000
000000000000

PA ⇒ 0001000000 0100000000
000000000000

Page directory offset = 1st 10 bits = 512

Page table offset = next 10 bits = 256

∴ The Value written at offset 256 in Pagetable (2nd level) should be the 1st 20 bits of PA + 12 flag bits

= 0001000000 0100000000 xxxxxxxxxx0x

At offset 512 in the page directory (1st level), we don't know 1st 20 bits (since we don't know the address of page table (2nd level)) but in the next 12 flag bits, R/W bit should be zero.

Q2

Page table reload

Q) Why is this zero ?

A) Because page tabe isn't setup and also pages aren't allocated

Q) How would we translate 0x80106c90 to a physical address?

A) By subtracting KERNBASE from VA (virtual to Physical (V2P) function)

Q) What is this?

A) Shifting 0x80106c90 Right by 22 bits tells us the 1st 10 bits which gives the index in 1st level of page table. kpgdir [0x200] gives the address of 2nd level page table page

Q) What does the 7 mean?

A) 7 implies that last 3 bits of entry are 1 in 0x3fe007.

*) Page is present, user is the Supervisor, page mode is writable

Q) What is this?

A) Offset 106 is the offset (index) in 2nd level of page table.

x106001 is the address value present at that index in 2nd level.

Q) Why 1 in the low bits?

A) It (1) implies that 0001 are last 4 bits.

User is user, page mode read

Page is present

Q) Why did the physical address work in gdb?

A) In cr3, Paging isn't enabled.
∴ Physical addresses can be directly provided.

Q) Why?

A) Paging gets enabled after switchkvm.
All addresses go through page table.
From now on, we need to provide Virtual addresses to it.

Q4 TRAPS

Q) Is it possible to have 2 context structures and One trapframe structure on kstack? If so, when? If not, Why not?

A) NO, there can't be 2 context to same kstack because, you have already switched by the time you push context. while pushing context, interrupt disabled.

Q) Is it possible to have 2 trapframe structures and 1 context structure on kstack. If so, when? if not, why not?

A) Yes, It will happen when there is interrupt in user process and runs kstack instruction. While running in kernel mode, timer interrupt occurs, then kernel stores another trapframe on the kstack & context switches to any other process. In these cases, there are 2 trapframe structures (system call -1, kernal mod interrupt -1) but has only one context structure

Q) Is it possible to have more than 3 sets of saved registers in kstack? If so, when? If not, why not?

A) NO. In XV6, there can't be more than 3 sets of saved registers bcoz there can be max of 2 trapframes and one context. It is ensured in XV6 by using locking and disabling interrupts. This prevents another interrupt from occurring when in middle of executing instructions of already occured interrupt.

Q5 Context Switching

Q) Where is the stack that sched() executes on?

A) It executes on the kernel stack of currently interrupted process.

Q) Where is the stack that scheduler() executes on?

A) On init process stack.

Q) When sched() calls swtch(), does that call to swtch() ever return? If so, when?

A) Yes, It returns when the currently interrupted process is once again scheduled.

Q) Could swtch do less work and still be correct? Could we reduce the size of a struct context?

A) NO. It is necessary to store that amount of data to maintain & follow correctness & calling conventions

Q) Four character pattern

A) c b a d

Q) The very 1st characters are ac. Why does this happen?

A) When system boots for the 1st time, the main fn calls the scheduler.. "a" gets printed. Next, when context switch occurs, sched gets called. ⇒ c

∴ a c