

Homework 7: Interrupts, threads and paging

Answer the following questions.

1. Give an example to demonstrate why precise interrupts are necessary? [2]
2. What is wrong with the following program (assuming this code compiles)? [1]

```
void bar(void *ptr) {
    int *arg = (int*)ptr;
    printf("value of x in foo is: %d\n", arg[0]);
    thread_exit();
}
```

```
void foo(void *ptr) {
    int x = 10;
    create_thread(bar, &x);
    thread_exit();
}
```

```
int main() {
    create_thread(foo, NULL);
    wait_for_all();
    return 0;
}
```

3. Show all intermediate steps during the creation of the page table of a process which is going to access virtual page numbers: 0x5001, 0x40010, 0x51011, and 0x30123. You have to use absolute addresses in all the steps. You can assume that the kernel virtual addresses in the range ‘0x80000000 - 0x80100000’ are free and managed by `alloc_page`. Whenever required, you can pick a free virtual page from this range. You can assume that the rest of the kernel code is not using `alloc_page`. The `KERNBASE`, `kva_to_va`, `va_to_pa`, and `alloc_page` are same as discussed in the class. Draw the final page table (page directory and page tables) with absolute values. [0.5]
4. Show all intermediate steps during the software walk of the page table to compute the physical address corresponding to virtual address 0x401000.

The `KERNBASE`, `kva_to_va`, and `va_to_pa` are the same as discussed in the class. Below are the details of the page table: [0.5]

- The `%cr3` register contains `0x500001`.
- The virtual address `0x80500004` contains `600007`.
- The virtual address `0x80600000` contains `800007`.
- The virtual address `0x80600004` contains `a00007`.
- The virtual address `0x80600008` contains `b00007`.

How to submit

Submit your handwritten homework in the submission box placed at the old academic building (2nd floor). The box will be placed on days when the homework is due.