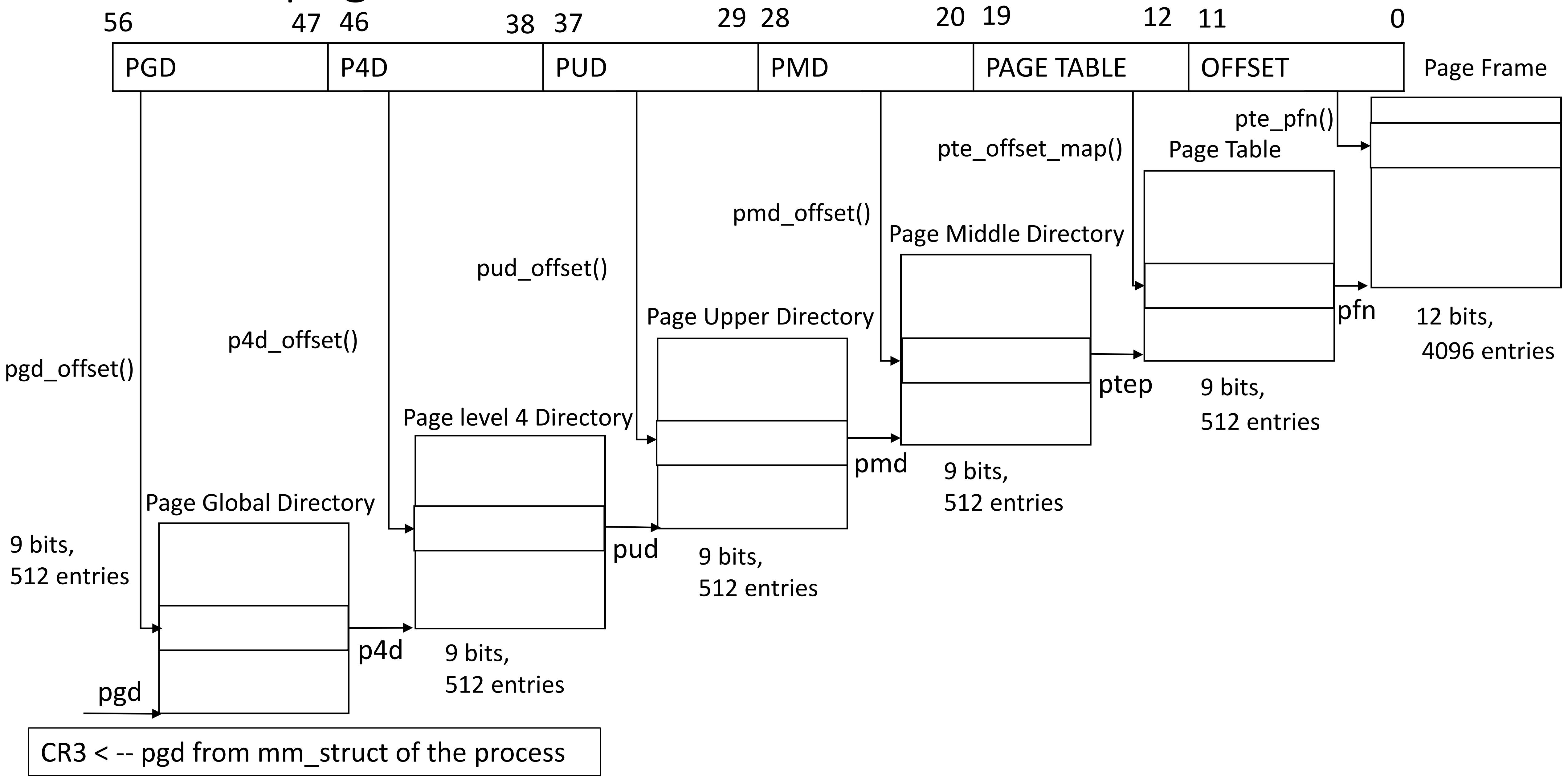
Assignment 3

CS550 – Operating Systems

5-level page table



56 – 47	46 – 38	37 – 29	28 – 20	19 – 12	11 – 0
PGD	P4D	PUD	PMD	PAGE TABLE	OFFSET

CR3 < -- pgd from mm_struct of the process

How to get physical frame number of page table?

- Get the base address of the page table – pgd
    - pgd_offset(mm, address);
- Every process is associated with task_struct structure that describes the process.
    - task_struct is defined in linux/sched.h and mm_struct is defined in mm_types.h

```
struct task_struct {
        …..
        struct mm_struct *mm;
        …..
};
struct mm_struct {
        …..
        pgd_t *pgd;
        …..
};
```

How to get physical frame number of page table?

- What do we know?
  - Process ID - passed as parameter to kernel module.
- How to get mm_struct using Process ID?

```
struct pid *pid;
struct task_struct *pid_struct;
struct mm_struct *pid_mm_struct;

pid = find_get_pid (process_ID);
pid_struct = pid_task(pid, PIDTYPE_PID);
pid_mm_struct = pid_struct->mm;

pgd = pgd_offset(pid_struct->mm, address);
```

How to get the virtual addresses of the process?

- Iterate through the virtual memory areas (VMAs)

```
for (vma = mm->mmap; vma; vma = vma->vm_next) {
        for(vaddr = vma->vm_start; vaddr < vma->vm_end; vaddr++){
                pgd = pgd_offset(mm, vaddr);
                …..
                …..
        }
}
```

Verify the VMA mappings using the command: `cat /proc/PID/smaps`