Assignment 3

CS550 – Operating Systems
5-level page table

<table>
<thead>
<tr>
<th></th>
<th>PGD</th>
<th>P4D</th>
<th>PUD</th>
<th>PMD</th>
<th>PAGE TABLE</th>
<th>OFFSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Page Frame</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- pgd_offset(): Page Global Directory
- p4d_offset(): Page level 4 Directory
- pud_offset(): Page Upper Directory
- pmd_offset(): Page Middle Directory
- pte_offset_map(): Page Table
- pte_pfn(): pfn

- 9 bits, 512 entries
- 9 bits, 512 entries
- 9 bits, 512 entries
- 9 bits, 512 entries
- 12 bits, 4096 entries

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)

```c
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`