Operating System

Project #2

16.10.10

Project Plan

- 5 projects
  - Install Xv6
  - System call + scheduling
  - Virtual memory (stack growth + COW)
  - Thread-support
  - Concurrency

- Single-handed project

Address Translation in Intel x86

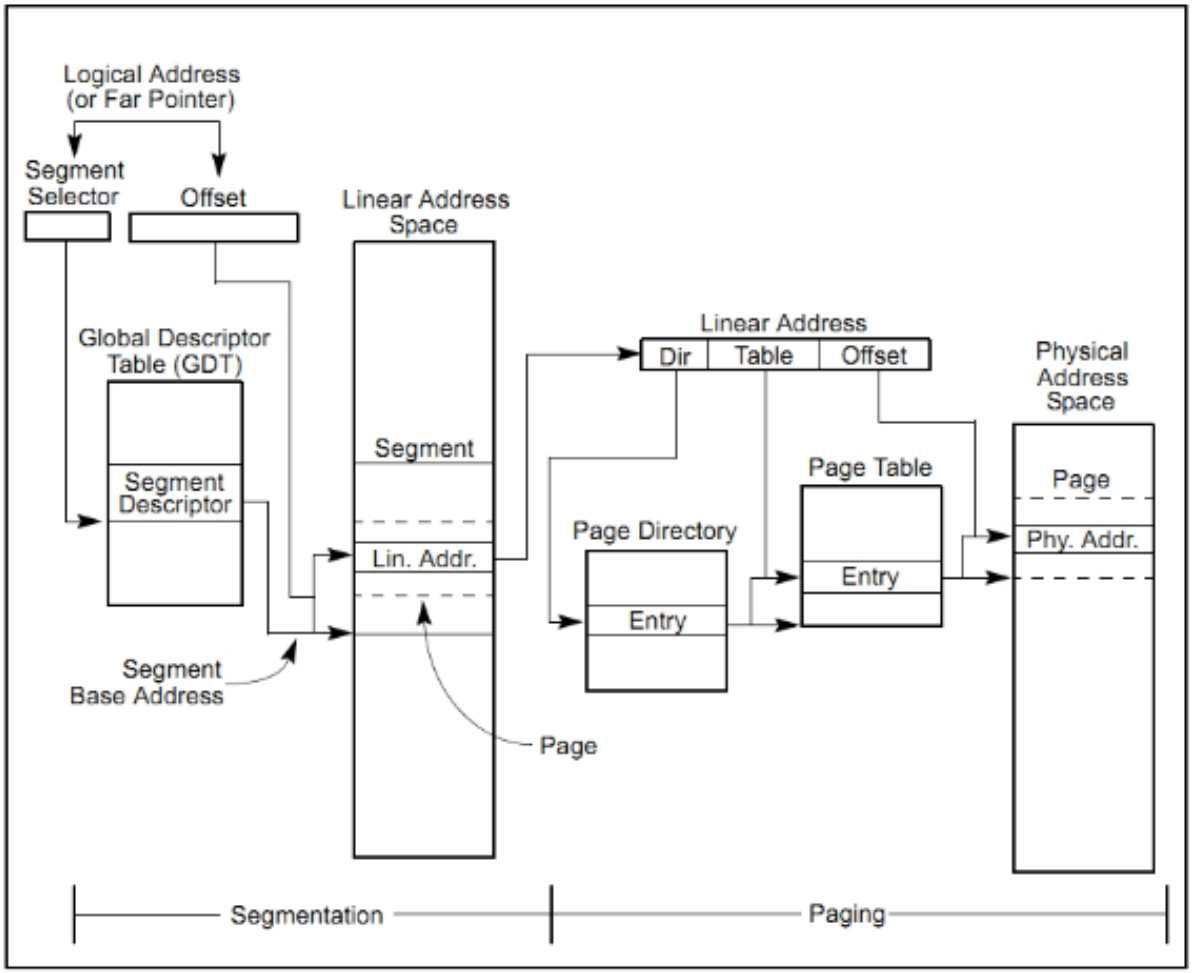
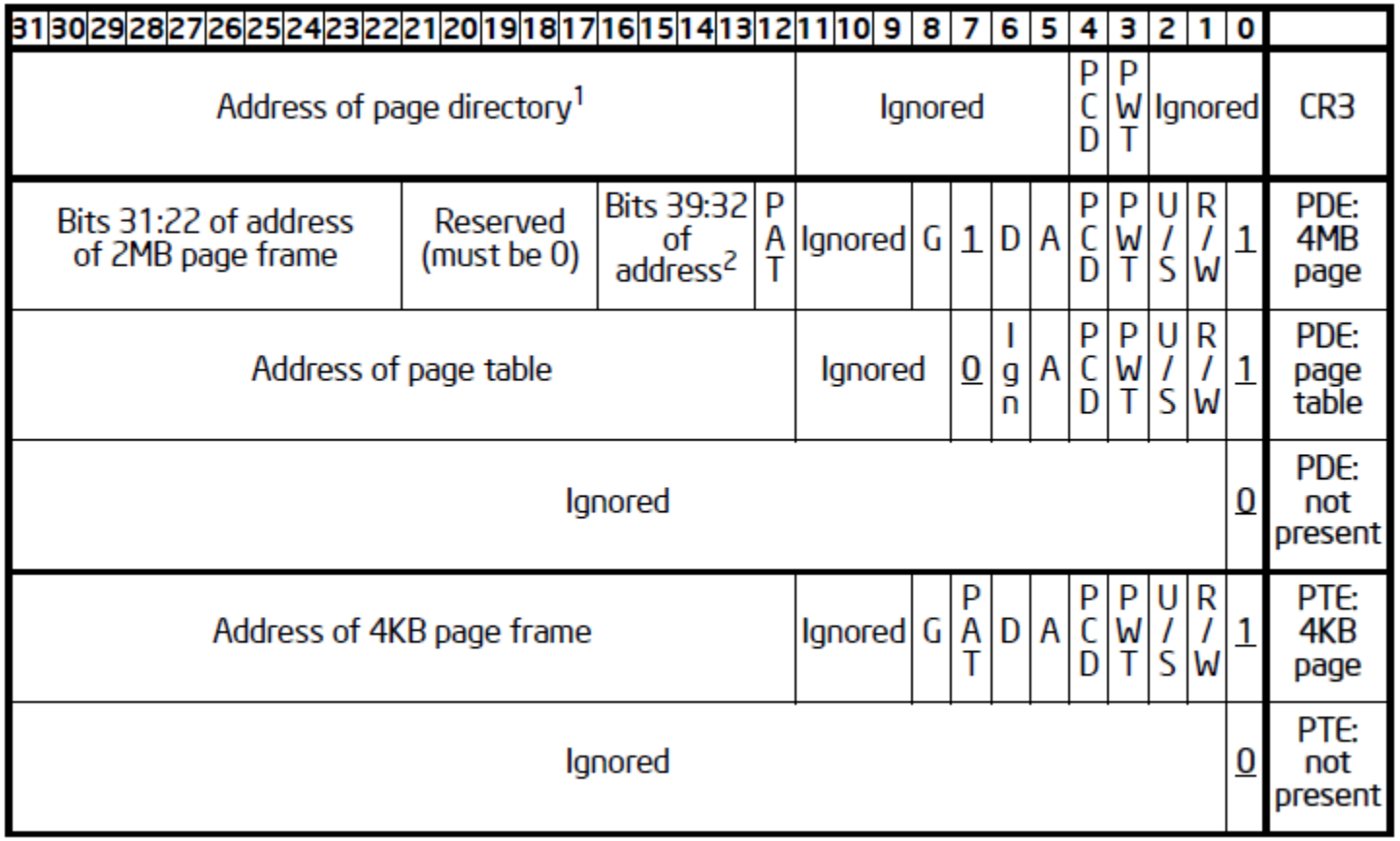


Figure 3-1. Segmentation and Paging

Formats of Paging Entries in Intel x86



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address of page directory[1]																				Ignored							PCD	PWT	Ignored			CR3
Bits 31:22 of address of 2MB page frame										Reserved (must be 0)						Bits 39:32 of address[2]			PAT	Ignored			G	1	D	A	PCD	PWT	U/S	R/W	1	PDE: 4MB page
Address of page table																				Ignored				0	Ign	A	PCD	PWT	U/S	R/W	1	PDE: page table
Ignored																															0	PDE: not present
Address of 4KB page frame																				Ignored			G	PAT	D	A	PCD	PWT	U/S	R/W	1	PTE: 4KB page
Ignored																															0	PTE: not present

Figure 4-4. Formats of CR3 and Paging-Structure Entries with 32-Bit Paging

Page Fault in Intel x86

- CR2 stores linear address that caused page fault
- Processor triggers interrupt #14 (page fault)

Control Registers in Intel x86

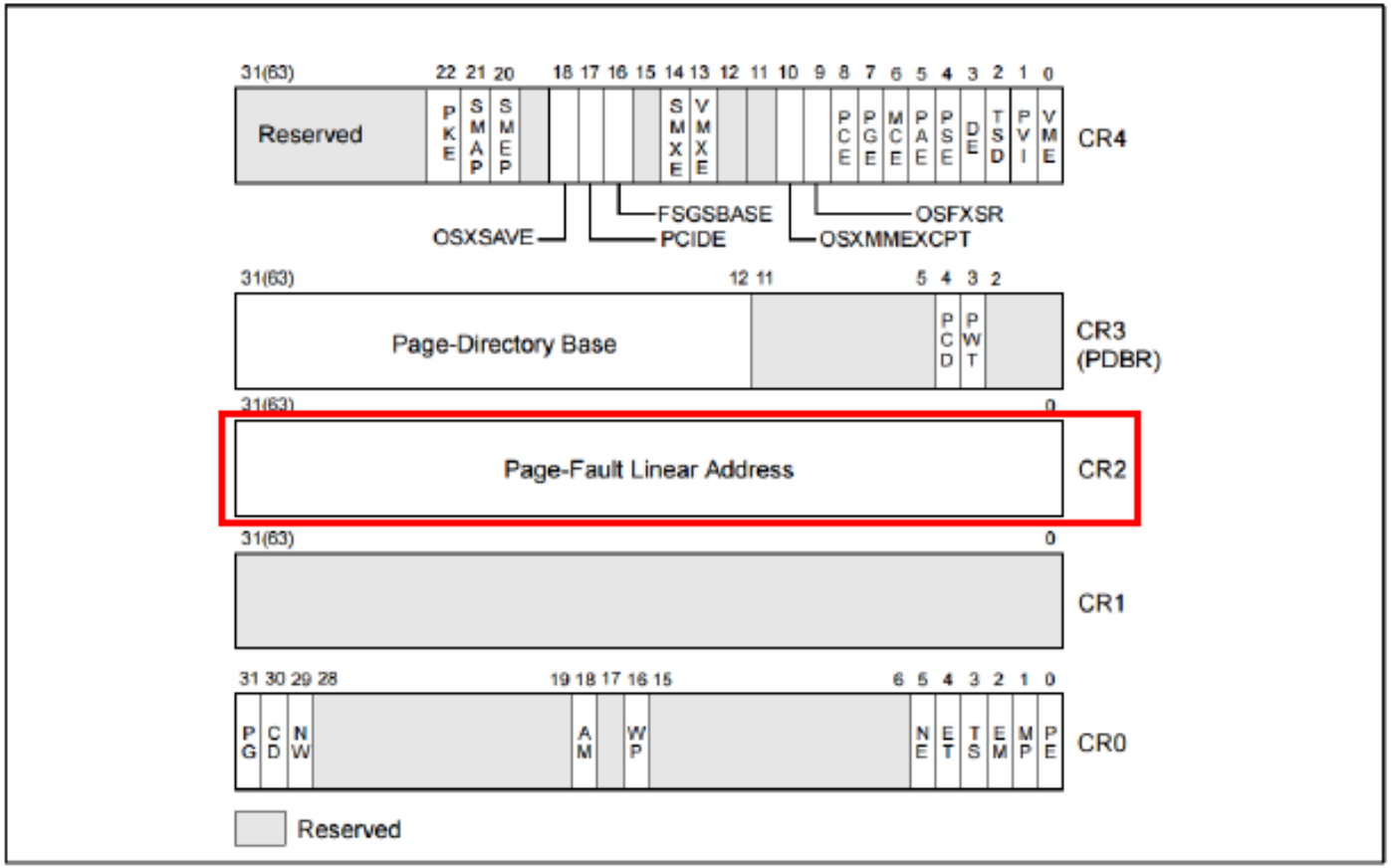


Figure 2-7. Control Registers

Exception & Interrupt Handling in xv6

- Follows Intel x86 architecture
- Procedures
  - Assign certain interrupt to interrupt descriptor table(IDT)
  - All interrupts jump to alltraps() and build trap frame
  - Handle each interrupt depending on its trap number
- Tip
  - Get page fault address with rcr2()

Virtual Address Space

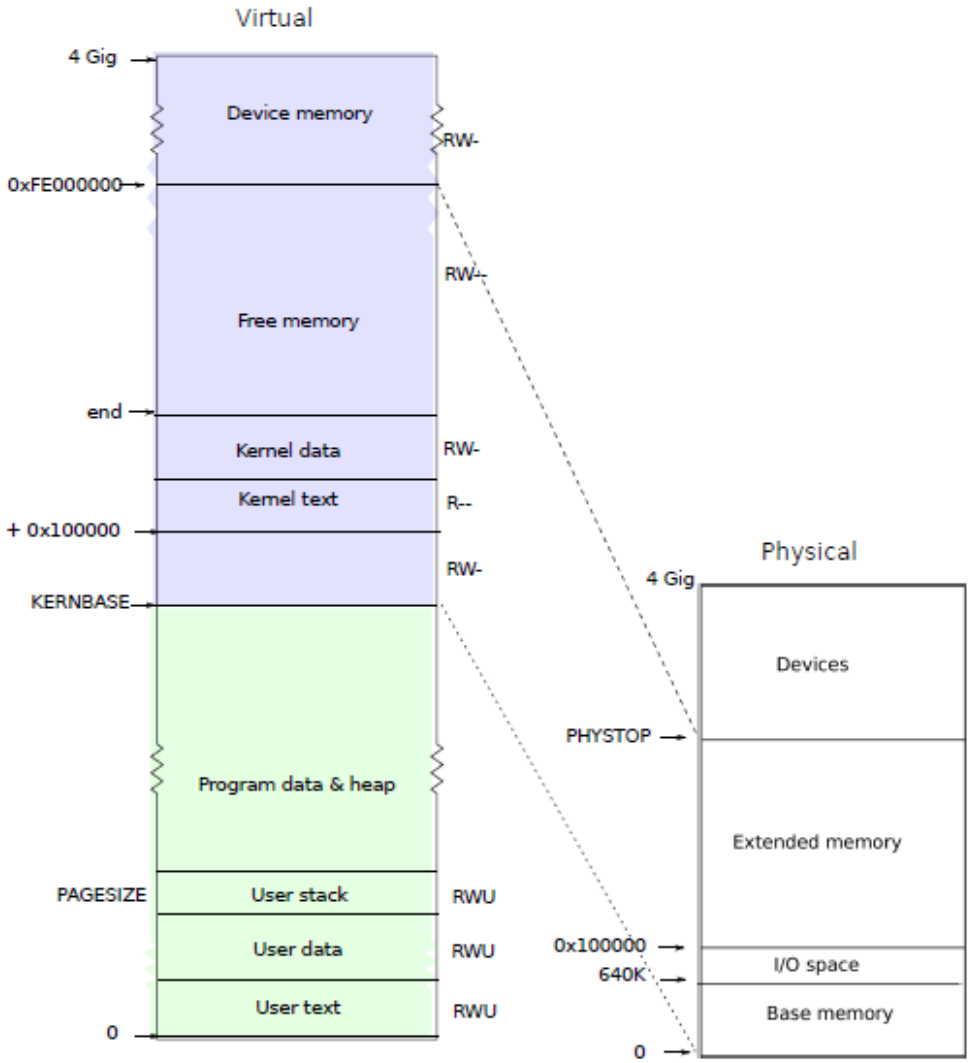


Figure 2-2. Layout of a virtual address space and the physical address space.

Process User Stack in xv6

- 1 stack page & 1 guard page

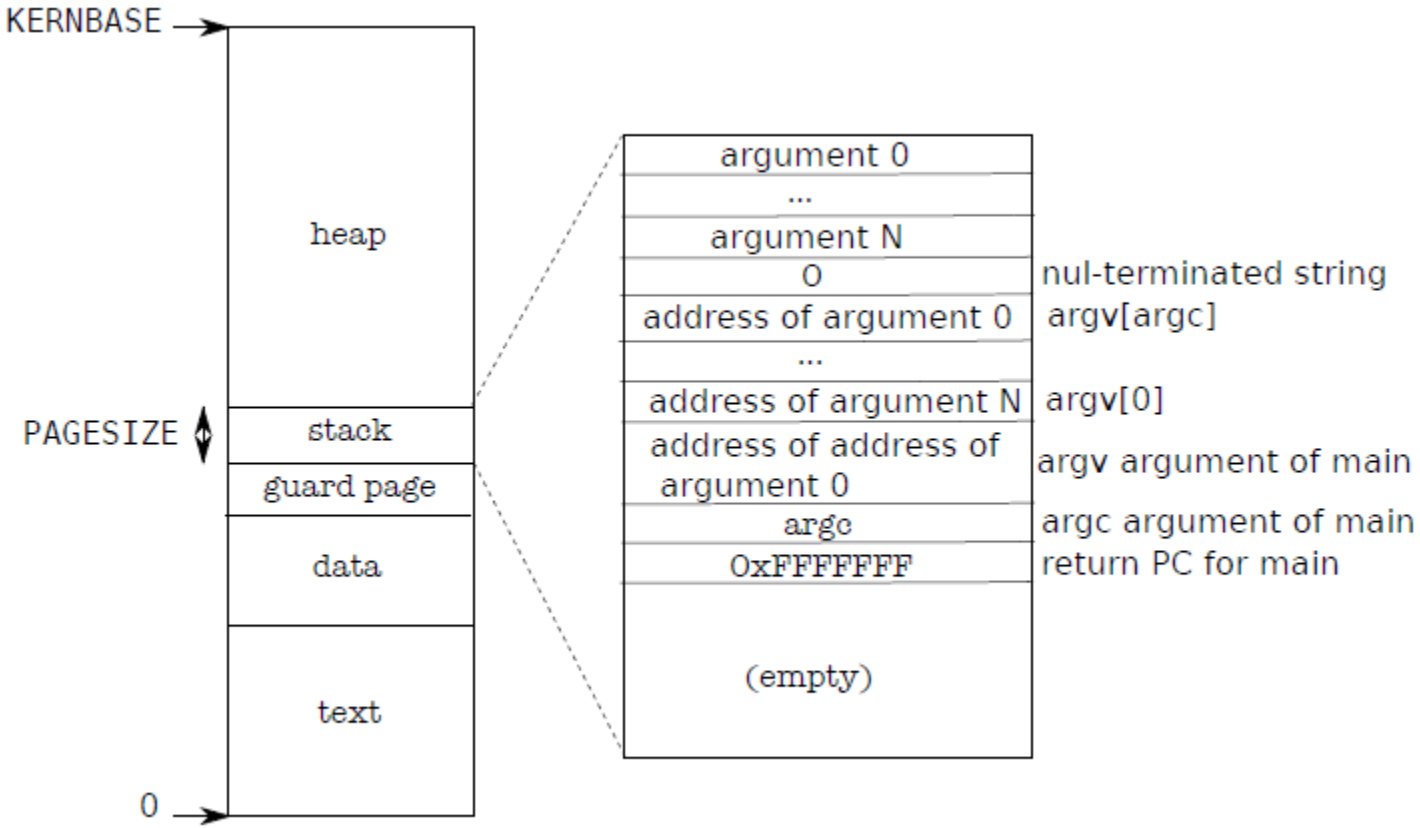
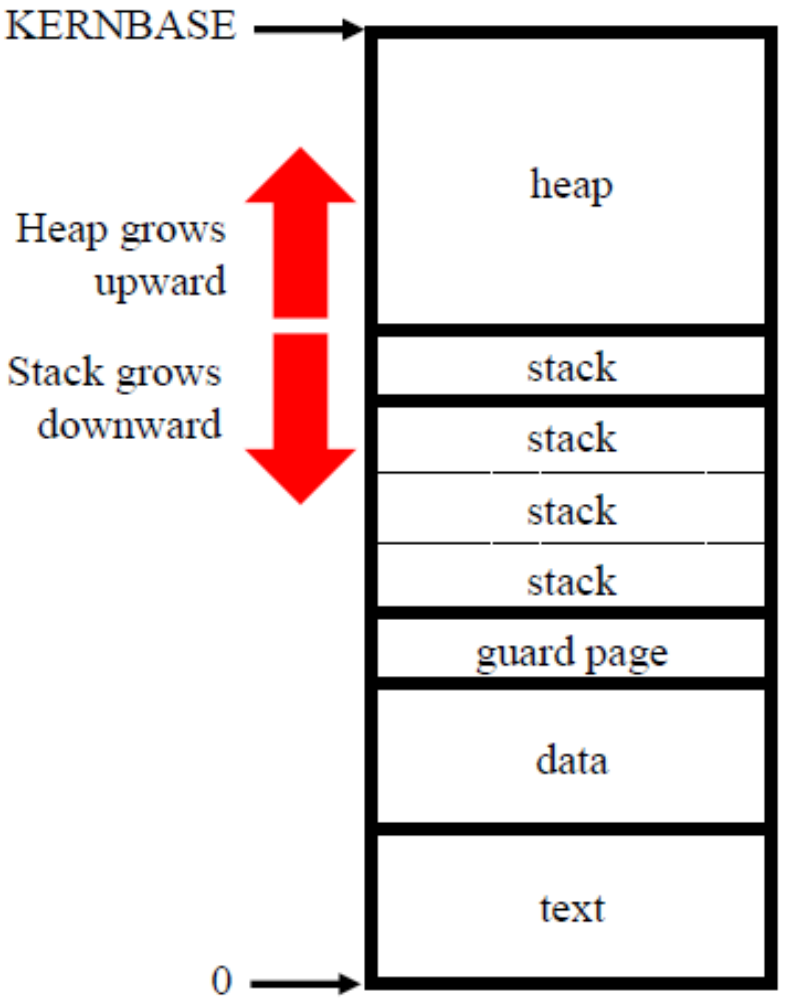


Figure 2-3. Memory layout of a user process with its initial stack.

Stack Growth in xv6

- 4 stack page & 1 guard page
- Stack grows when current stack is full
- Stack pointer can move up to 32bytes (pushal instruction)
  - trapframe->esp
- When stack pointer reaches guard page, stack overflow occurs and process is killed



Project #2 – Stack Growth

- Implement stack growth in xv6
- Submit a tar.gz file
- Send email to T.A
  - [SSE3044]Project#2-YOURID-YOURNAME
    - ex) [SSE3044]Project#2-2016710580-이규선
  - Email address : lgs0409@naver.com
  - Wrong title is not allowed
- Due date
  - 2016-10-30(Sun) PM 23:59
  - Get no point for late submission