Paging

CS143A: Principles of operating systems - Fall'17

UC Irvine, California

- Remember designated initializers?

Designated initializer

- Remember designated initializers?
- ```
  static uchar togglecode[256] = {
    [0x3A] CAPSLOCK,
    [0x45] NUMLOCK,
  };
  ```

Designated initializer

- Remember designated initializers?

- ```
  static uchar togglecode[256] = {
    [0x3A] CAPSLOCK,
    [0x45] NUMLOCK,
  };
  ```

- ```
  togglecode[0x3A] = CAPSLOCK;
  togglecode[0x45] = NUMLOCK;
  ```

Normal pages (4KiB pages)

31	22	21	12	11	0
PDE		**PTE**		Page offset	
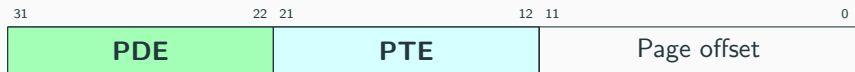
- PDE - ($2^{10} = 1024$ entries)
- PTE - ($2^{10} = 1024$ entries)
- Page offset - ($2^{12} = 4KiB$)

Page Size Extension (4MiB pages)

31 22	21 0
PDE	Page offset

- PDE - ($2^{10} = 1024$ entries)
- Page offset - ($2^{22} = 4MiB$)

xv6 Boot/Initial page table

- Initial page table (main.c)
  ```
  pde_t entrypgdir[NPDENTRIES] = {
    // Entry 0
    [0] = (0) | PTE_P | PTE_W | PTE_PS,
    // Entry 512
    [KERNBASE>>PDXSHIFT] = (0) | PTE_P | PTE_W | PTE_PS,
  };
  ```

xv6 Boot/Initial page table

- Initial page table (main.c)
  ```
  pde_t entrypgdir[NPDENTRIES] = {
    // Entry 0
    [0] = (0) | PTE_P | PTE_W | PTE_PS,
    // Entry 512
    [KERNBASE>>PDXSHIFT] = (0) | PTE_P | PTE_W | PTE_PS,
  };
  ```
- PTE_P - Page present

xv6 Boot/Initial page table

- Initial page table (main.c)
  ```
  pde_t entrypgdir[NPDENTRIES] = {
    // Entry 0
    [0] = (0) | PTE_P | PTE_W | PTE_PS,
    // Entry 512
    [KERNBASE>>PDXSHIFT] = (0) | PTE_P | PTE_W | PTE_PS,
  };
  ```
- PTE_P - Page present
- PTE_PS - 4MiB pages

xv6 Boot/Initial page table

- Initial page table (main.c)
  ```
  pde_t entrypgdir[NPDENTRIES] = {
    // Entry 0
    [0] = (0) | PTE_P | PTE_W | PTE_PS,
    // Entry 512
    [KERNBASE>>PDXSHIFT] = (0) | PTE_P | PTE_W | PTE_PS,
  };
  ```
- PTE_P - Page present
- PTE_PS - 4MiB pages
- Two virtual 4MiB pages point to the same physical page
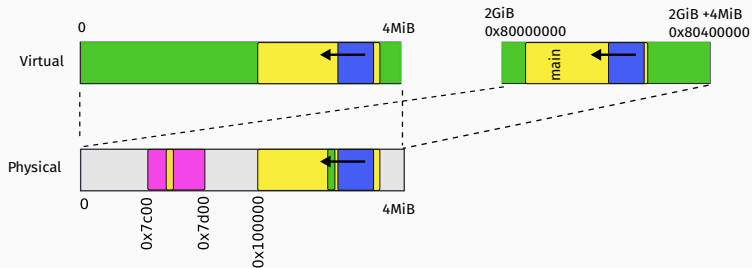
- Program CR3

```
# Set page directory
movl    $(V2P_WO(entrypgdir)), %eax
movl    %eax, %cr3
```

---

[1]entry.S

- Program CR3

```asm
# Set page directory
movl    $(V2P_WO(entrypgdir)), %eax
movl    %eax, %cr3
```

- Program CR0

```asm
movl    %cr0, %eax
orl     $(CR0_PG|CR0_WP), %eax
movl    %eax, %cr0
```

- Program CR3

```
# Set page directory
movl    $(V2P_WO(entrypgdir)), %eax
movl    %eax, %cr3
```

- Program CR0

```
movl    %cr0, %eax
orl     $(CR0_PG|CR0_WP), %eax
movl    %eax, %cr0
```

- CR0_PG - Paging - enable paging and use the CR3 register

---

[1]entry.S

Source tree

- Kernel objects

  ```
  bio.o console.o exec.o  file.o fs.o ide.o ioapic.o kalloc.o kbd.o
  ↪   lapic.o log.o main.o mp.o picirq.o pipe.o proc.o sleeplock.o
  ↪   spinlock.o string.o swtch.o syscall.o sysfile.o sysproc.o
  ↪   trapasm.o trap.o uart.o vectors.o vm.o
  ```

Source tree

- Kernel objects

  ```
  bio.o console.o exec.o  file.o fs.o ide.o ioapic.o kalloc.o kbd.o
  ↪   lapic.o log.o main.o mp.o picirq.o pipe.o proc.o sleeplock.o
  ↪   spinlock.o string.o swtch.o syscall.o sysfile.o sysproc.o
  ↪   trapasm.o trap.o uart.o vectors.o vm.o
  ```

- User applications

  ```
  cat.o echo.o forktest.o grep.o init.o kill.o ln.o ls.o mkdir.o rm.o
  ↪   sh.o stressfs.o usertests.o wc.o zombie.o hello.o
  ```

Source tree

- Kernel objects

  ```
  bio.o console.o exec.o  file.o fs.o ide.o ioapic.o kalloc.o kbd.o
  ↪  lapic.o log.o main.o mp.o picirq.o pipe.o proc.o sleeplock.o
  ↪  spinlock.o string.o swtch.o syscall.o sysfile.o sysproc.o
  ↪  trapasm.o trap.o uart.o vectors.o vm.o
  ```

- User applications

  ```
  cat.o echo.o forktest.o grep.o init.o kill.o ln.o ls.o mkdir.o rm.o
  ↪  sh.o stressfs.o usertests.o wc.o zombie.o hello.o
  ```

- Low-level assembly

  ```
  bootasm.S entry.S swtch.S usys.S entryother.S initcode.S trapasm.S
  ```

Source tree

- Kernel objects

  bio.o console.o exec.o file.o fs.o ide.o ioapic.o kalloc.o kbd.o
  ↪   lapic.o log.o main.o mp.o picirq.o pipe.o proc.o sleeplock.o
  ↪   spinlock.o string.o swtch.o syscall.o sysfile.o sysproc.o
  ↪   trapasm.o trap.o uart.o vectors.o vm.o

- User applications

  cat.o echo.o forktest.o grep.o init.o kill.o ln.o ls.o mkdir.o rm.o
  ↪   sh.o stressfs.o usertests.o wc.o zombie.o hello.o

- Low-level assembly

  bootasm.S entry.S swtch.S usys.S entryother.S initcode.S trapasm.S

- Header files, Helper scripts

- Makefile

User space applications

- Programs

```
UPROGS=\
        _cat\
        _echo\
        ...
```

- Programs
  ```
  UPROGS=\
          _cat\
          _echo\
          ...
  ```

- Rule (Pattern match[2] and built-in rules[3])
  ```
  # User space helper libs
  ULIB = ulib.o usys.o printf.o umalloc.o

  _%: %.o $(ULIB)
          $(LD) $(LDFLAGS) -N -e main -Ttext 0 -o $@ $^
  ```

---

[2]https://www.gnu.org/software/make/manual/html_node/Pattern-Rules.html
[3]https:
//www.gnu.org/software/make/manual/html_node/Catalogue-of-Rules.html