MP 3.3 Overview

Okan K., Pauline L.

Read the Docs & Code CAREFULLY!!

Before we start

1. Read Docs
2. Read MP3 tips
3. Read the code
4. Read the Appendices
5. Read Intel manual
6. Read the descriptors PDF (Tools, References, and Links on website)
7. If you still can’t figure out come to OH

Parts

- General Syscalls
- Syscall linkage
- Structs(pcb)
- Execute
- Halt

Structs

- Pcb_t:
  - Pid
  - Parent_id
  - File descriptor
  - Saved_esp
  - saved _ebp
  - active

Execute

Execute

1. Paging Helpers (optional, but very helpful)
   - Map Virtual & Physical Memory (optional, needed for CP5)
   - Unmap Virtual & Physical Memory (optional, needed for CP5)
2. Parse cmd
3. File Checks
4. Create new PCB
5. Setup memory (aka paging)
6. Read exe data
7. Setup old stack & eip
8. Goto usermode

Execute-Paging

- Mapping Phys to Virtual func
- Unmap(optional)

Execute-File Checks

- Does the file exist?
- Is the file an EXE?
  - Hint: look in MP3 documentation for some “magic numbers”
- Is the file valid?
- Remember to get prog_eip from valid files:
  - Look in MP3 documentation for how to do this

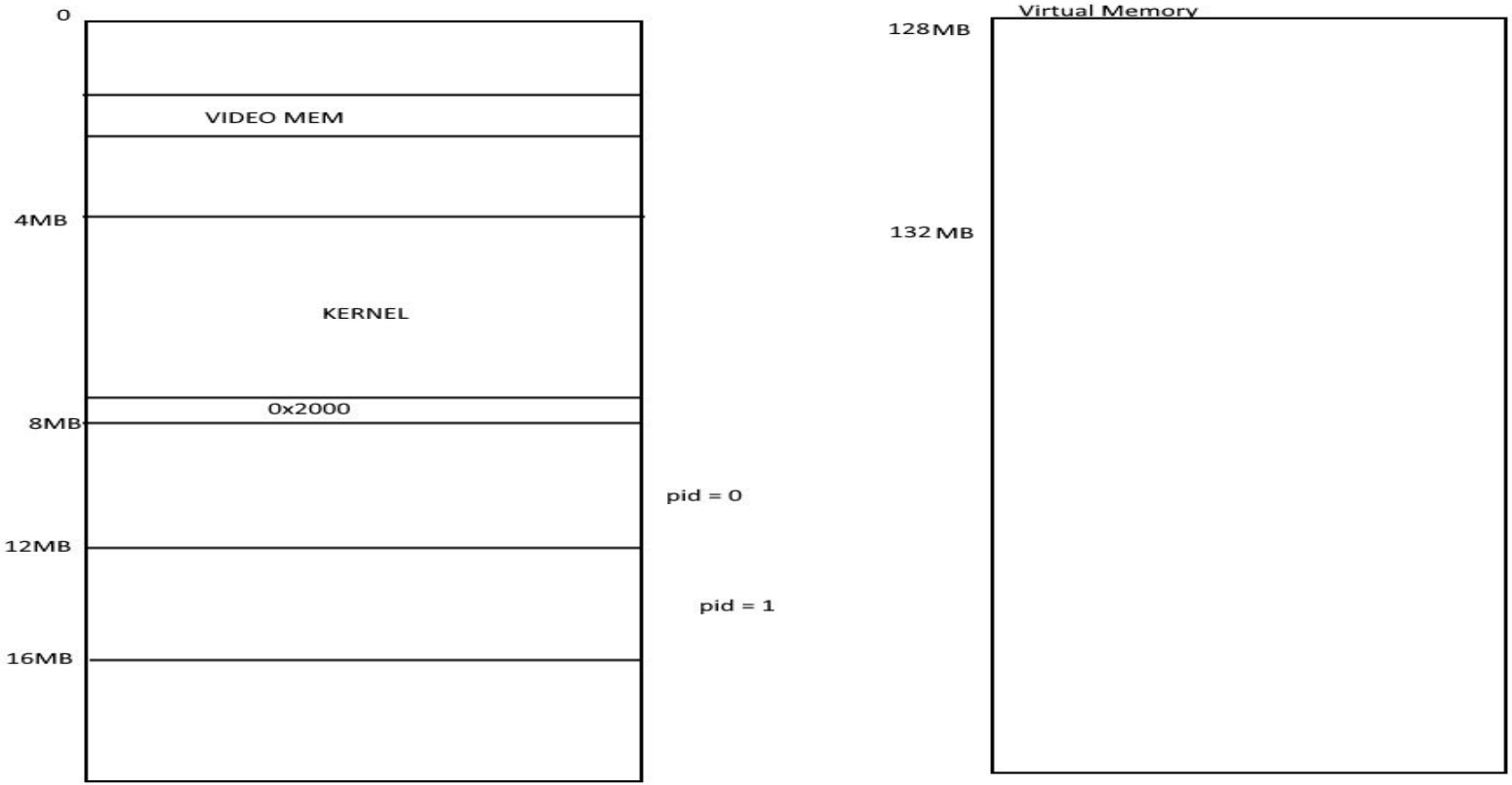
Execute-Create PCB

- Give pcb memory
- Set active
- Set file descriptor

Execute-Setup Memory

- Setup paging(you should use some helper functions)
  - REMEMBER TO FLUSH TLB (OSDev is very helpful for this)
- Set save_ebp
- Set save_esp (to a fixed value)

Execute-Setup Memory



Execute-Save old Stack

- Hint:
  - You can save current esp and ebp values by:

```
register uint32_t saved_ebp asm("ebp");
```

  - Or you can just use normal inline asm

Execute-Usermode

- OSdev Getting_to_Ring_3
- Switching to usermode requires pushing certain things to stack
  - user_ds
  - user_esp
  - user_cs
  - prog_eip
- Look in MP3 Documentation for some of the above values

Halt

Halt-Parts

- Setup return value:
  - Check if exception
  - Check if program finished
- Close all processes
- Set currently-active-process to non-active
- Check if main shell
  - Restart main shell
- Not main shell handler (cntd.)
- Halt return (asm)

Halt-Handling non-main handler

- Get parent process
- Set tss for parent
- Unmap paging for current-process
- Map parent’s pagining
- Set parent’s process as active
- Call halt return (asm)

Halt-Halt return

- Take in esp,ebp and return value
- Set esp,ebp register as esp,ebp arguments
- Set eax register as the return value

Bugs

Common Bugs-Page fault

- You setup memory in Execute incorrectly
- You setup memory in Halt incorrectly
- Your esp, ebp values are wrong
- You fill up file descriptor wrong

Common Bugs-General Protection (GPE)

- Your esp, ebp values are wrong
- Your eip is wrong
- Your goto usermode is wrong
- Your halt return helper is wrong

Common Bugs-Double or Triple Fault

- You are getting a Page fault inside your GPE or vice versa
- Refer to previous slides to debug