ECE 391 Discussion Week 10

Announcements & Reminders

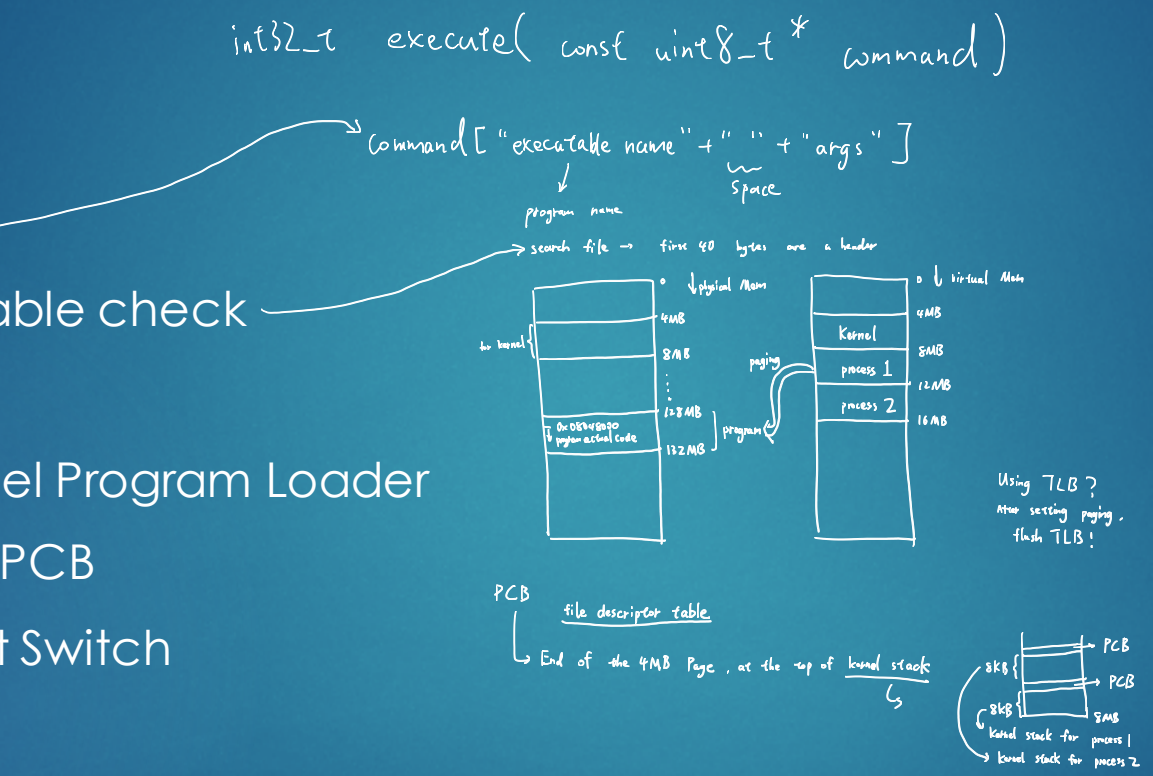
- Exam 2 is next Tuesday, November 6th at 7pm
  - Room assignments TBA
  - TWO sheets of notes (front and back). No devices/calculators!
  - Conflict? Notify the Professors by Friday, November 2nd at 5pm
  - Review session TBD
- MP3.3 due November 12th at 6pm
- Plan accordingly!

System Calls

- System calls
  - Support 10 (numbered 1-10)
  - CP3: Execute, halt, open, close, read, write
  - CP4: The rest
- Parameters (convention)
  - EAX – system call number
  - EBX – 1st arg
  - ECX – 2nd arg
  - EDX – 3rd arg
  - Return value in EAX (not all calls return)
- All system calls must be supported via a common IDT entry → Need generic assembly linkage

Execute

- Execute
  - Parse
  - Executable check
  - Paging
  - User-level Program Loader
  - Create PCB
  - Context Switch

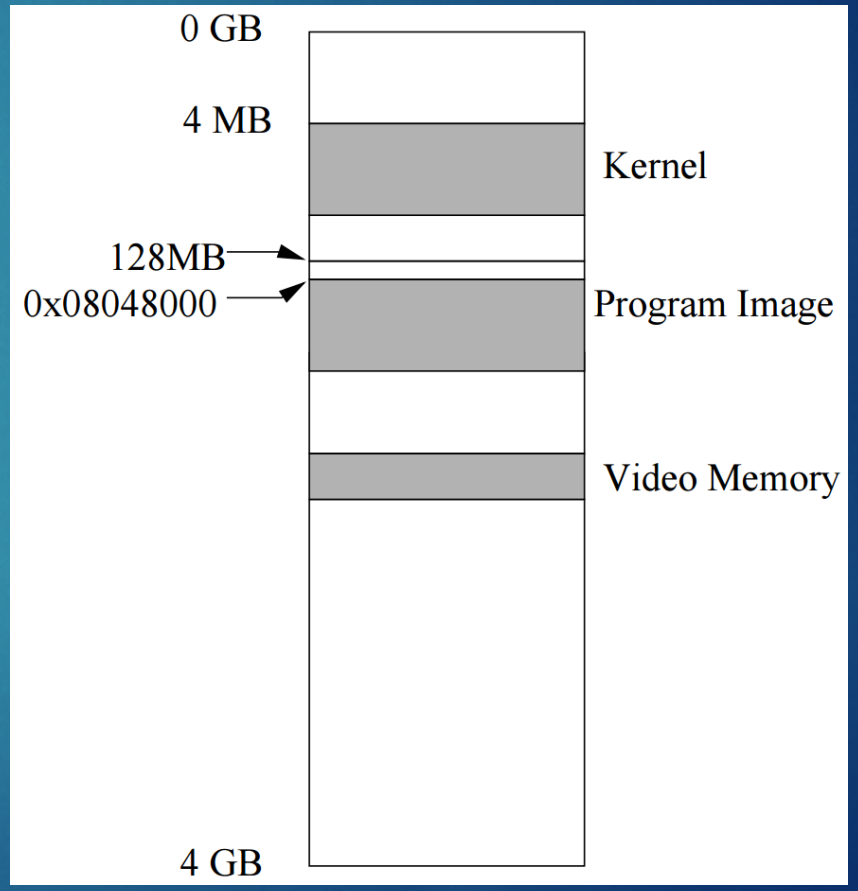


Paging and Program Loader

- All user level programs will be loaded in the page starting at 128MB (virtual mem)
- Physical memory starts at

  8MB + (process number * 4MB)

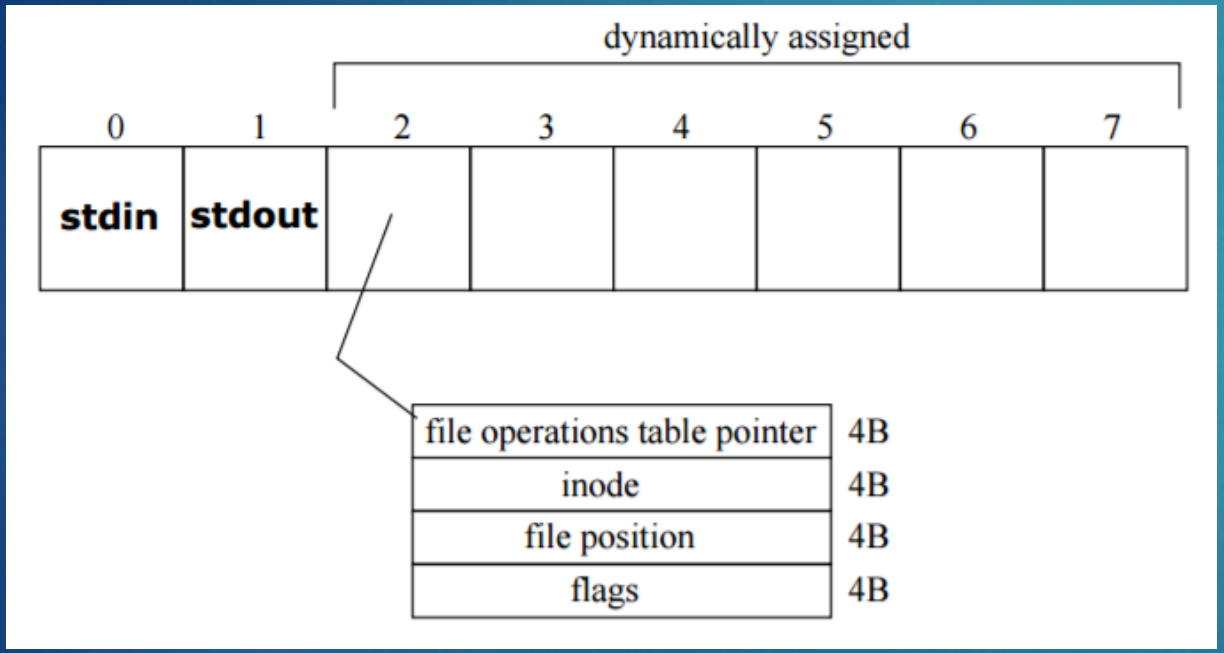
  process number starts from 0
- Flush TLB after swapping page
- Program loader
  - Checks ELF magic constant
  - Copy file contents to the correct location
  - Find the first instruction's address



Process Control Block (PCB)

- Each process has its own PCB
- Each PCB starts at the top of a 8KB block inside the kernel
  - First PCB starts at 8MB – 8KB
  - Second PCB starts at 8MB – 8KB – 8KB
  - …
- Switch the kernel stack when switching PCB
- What you need in a PCB
  - File descriptor array
  - Whatever you think is necessary for running/stopping a program and should be process specific

File Descriptor Table



- File descriptor
  - integer index into this array
  - how user programs identify the open file
- Up to 8 open files per task
- Stores structures containing:
  - Pointer to file operations jump table
  - Inode
  - File position
  - Flags

System Calls (contd.)

- Open
  - Find the file in the file system and assign an unused file descriptor
  - File descriptors need to be set up according to the file type
- Close
  - Close the file descriptor passed in (set it to be available)
  - Check for invalid descriptors
- Read, write
  - Use file operations jump table to call the corresponding read or write function
  - The real functionality should be no longer than 1 line

Now we combine everything