Lecture 2

MS-Dos vs. Unix / Reading Review

C vs. asm

- Dos: did not have in-house compiler
- Multi-programming
  - Bell naturally assumed many terminals to one machine
  - Computing is a "utility"
  - PC <> "Single User Machine"
- Kernel & Shell vs. OS & Program
  - Layer of abstraction in Kernel & Shell vs. OS & Program
- Hard drive vs. Floppy
- Unix: Everything is a file
  - Devices as file - generic file I/O
  - Pipes & I/O redirecction
  - Separated file name from the actual file
- Security
  - P.C. was significantly less secure
Cmd line processor:
loop:
  check T1
  if not add to but 1
  if newline
  run prog 1
  T2 ...
  T3 ...
  T4 ...

We switch context by switching stacks

P1

P2

OS state: 

Stack:

switch
push R0...R7
P1.stack = SP
SP = P2.stack
pop R7...R0
return

Stack:

P1 3

X -> switch()
GETCHAR()  
if T1. ready
   Switch P1
if T2. ready
   Switch P2

y → CALL x  
PUSH PC
JMP x

CPU

Timer_IRQ_handler:  
switch O  
→ Time-Slicing
Need to add a supervisor process

User State:
  only modify R0...R7, PC, SP

Supervisor State:
  full access

switch via an exception