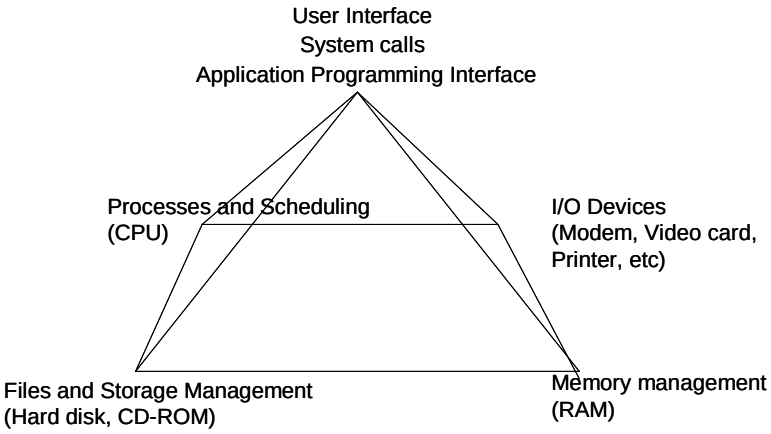
SINGAPORE POLYTECHNIC
SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING
ET0023 OPERATING SYSTEMS

TUTORIAL 2 – Computer Systems Review (Suggested Solutions)

1.

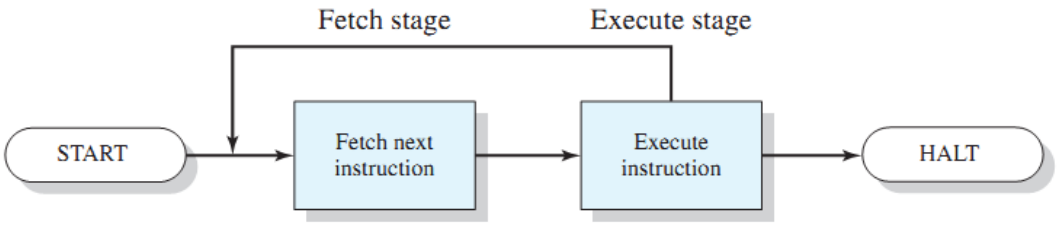


Similarities	Desktop OS	Server OS
Processes & Scheduling	Gives priority to visible (or top) operations which requires user interaction	Gives more priority to background jobs as it runs these as daemons
Video operations	Gives priority to video operations through user interaction	Usually has poor or no GUI since there are multiple users
Memory Management	Gives the current running program as much memory as required to affect smoother operations	Each job has to apply for the amount of memory and is allocated just enough, as multiple jobs have priority
File and Storage Management	May cache or buffer disk operations in order to facilitate user activity as higher priority	Disk operations are given high priority so as to clear the storage queue for multiple operations

2. Ref: http://en.wikipedia.org/wiki/Intel_8086

Register	Example	Purpose
General Registers	AX, BX, CX, DX	Temp storage locations for intermediate calculations
Program Counter	Instruction Pointer (IP)	Keeps track of the NEXT instruction to be executed
Stack Pointer	SP	Points to top of the Stack memory area. The stack memory area allows data to be PUSHed or PULled automatically using SP which increments/decrements accordingly
Program Status Word	Status Register	Keeps track of the arithmetic and logical operations by setting/clearing bits indicating the status of the operation e.g. ZF=1 when operation results in a 0

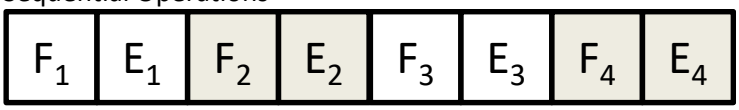
3.



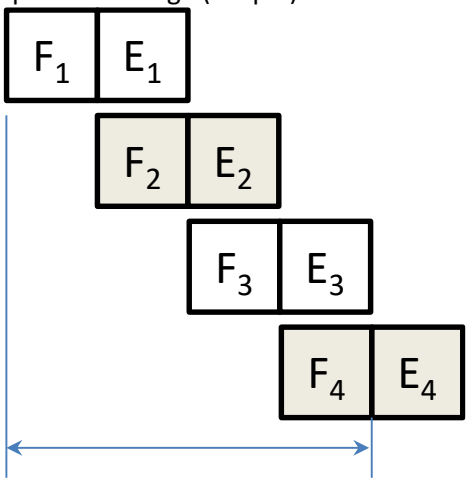
(From current lecture transparencies)

http://en.wikipedia.org/wiki/Instruction_pipeline

A normal CPU operation would be Fetch-Execute-Fetch-Execute,
Sequential Operations



Pipelined 4-stage (simple)



Shortened time, as the Fetch overlaps with the execute stages

Problems arise when
a) One of the instructions is a JUMP/BRA (i.e. transfer of control)
b) Another problem is when the Execute involves a WRITE which needs to use the Buses

What happens then? Flush the queue, delay the instructions?
Get the students to simulate the operations in class

4. API – Application Programming Interface
(http://en.wikipedia.org/wiki/Application_programming_interface)
What is an API and what is it good for?
(http://www.makeuseof.com/tag/api-good-technology-explained/)

How APIs can help development of software:
a) Routines and libraries already built for you
b) Standard method of developing software which gives standard output

How APIs fail in the development of software

a) Cannot develop software for particular hardware e.g. video cards for specialised operations
b) Can only develop for specific hardware platforms
c) Must conform to the requirements of the API e.g. Apple IOS

5.

a) MS-DOS is a single user, CLI, built of x86 processors. It allows users to interact with systems of a common hardware base i.e. x86 and provide necessary functions for Storage, Memory Management, peripheral management and running of programs. It could not, however, be used to run GUIs unless specific drivers/libraries were written for it. MS-DOS run in the real-mode of the x86, having access to only a maximum of 1 GB of RAM

b) MS-DOS is no longer in use today. It is superseded by the CLI of Windows OS and other modern day real-mode OS.

c) FreeDOS (http://www.freedos.org/)
An Open-source Real-mode operating system which is commonly used/distributed for updating flash ROMs on systems, repairing un-bootable systems and performing hardware maintenance.

d) Use the MS-DOS API

```
MOV   DL, "A"        ; output the character A
MOV   AH, 2          ;MSDOS API Code for function 2, write to STDOUT
INT   21H            ;execute via API
```

There is no need to understand the hardware or setup of the underlying system.
Will work on any system running MS-DOS