Hw 1. Chapter 1

5, 6, 7, 8, 9, 10, 12, 13

- typo on #9 $106 = 10^6$

1.6

@ ① first check for input flag FGI

② if FGI == 0, data is accepted by input register INPR.

③ upon arriving 8-bit data from teletype it is stored in input register INPR.

④ FGI is periodically checked and if "FGI = 1"

1.5 pg 38

32 bit cpu

16 bit bus

8 Mylz clock, 8 mil clocks a second

8 mil / 4 = 2 mil memory cycle

4 mil bytes a second

⑥ double bus width to match register size

1.6

1.7 DMA is doing actual I/O, because it is transferring data, because there is a process waiting for the I/O data

1.8 · 9600 bps modem @ 1mil instructions per second

· we need to know bus width assume 32-bit.

$\frac{9600}{32} = 300$ cycles/s

1.9 5% used for I/O transfers

$5\% \cdot (10^6 \text{ instructions}) = 50{,}000 \frac{\text{instructions}}{\text{second}}$ for I/O

25,000 words.

$0.05 * 10^6 * 5 = 250{,}000$ memory cycles

$0.95 * 10^6 * 2 = 1{,}900{,}000$ memory cycles.

total of 2,150,000 memory cycles

upper bound

1.10 consider the code

```
for (i=0; i<20; i++)
    for (j=0; j<10; j++)
        a[i] = a[i] * j      or   a[i, j] = 0
```

locality principle

- memory & programs tend to stay local.

1.12 $T_C = 100ns$ $C_C = 0.01$ cents/bit

$T_M = 1,200ns$ $C_M = 0.001$ cents/bit

$$12 \text{ mil} \cdot 8(0.001) =$$

$T = 110 ns = \text{Hit ratio } H \; 100ns + (1-H)(1200ns + 100ns)$

$110 = 100H + 1300 - 1300H \qquad 1300 ns$

$1200H = 1190$

$$H = \frac{1190}{1200} = .991 \approx 99\%$$

1.13 T_C = time cash - 20ns $H_C = .9$ cache

T_M = time memory - 60ns $H_M = .6$ memory

$T_{disk} = 12 ms$

$T = .9(20ns) + .1(0.6(60) + (0.4)(12 ms = 12,000,000ns)$

$= 18 + 4.8 + 480,003.2$

= 480,026. ns effective access time.

1.H

8/29 Day 5

Continued from last thing week 1

Performance Goals of OS

1. Provide high utilization of resources to keep costs down
2. Should provide high availability of resources
3. Promote user productivity
   - Make sure user works well
4. Reliable
5. Provide protection for important resources.
6. efficienty
   - Operating system needs small footprint.
7. predictable
   - Action should produce same result each time.
8. easy to use
9. Provide system resources
   - User needs enough CPU / memory / Access to I/O
   - Access to managment of system resources
     communication between programs
   - Provide device independence
     - Plug & play
     - Access to files.
   - diagnostic facilities

Note: Core dump is the file of contents of ram

Types of operating systems

1. Batch systems - Like a payroll application
   - for Batch processing
   - all data present before program is run
   - no interaction while program runs
   - no results until finished.
   - Goal
     - minimize turnaround time
     - maximize throughput
   - Current Example
     - virus scan
     - updates

2. time Sharing / interactive system
   - use time slicing to share CPU
   - Goal
     - Minimize response time
   - interaction is allowed while program is running.

3. transaction process system
   - characterized by a rigid controlled dialog between user and app.
   - Goal

- no incomplete transactions
- reliable response time

④ Real time systems - Motor controlls
- Proces events that occur in "real" world.
- not interactive
- built for one purpose
- Goal is to meet deadline.
- Soft real time
- hard real time

⑤ Personal computer system
- Generally one user

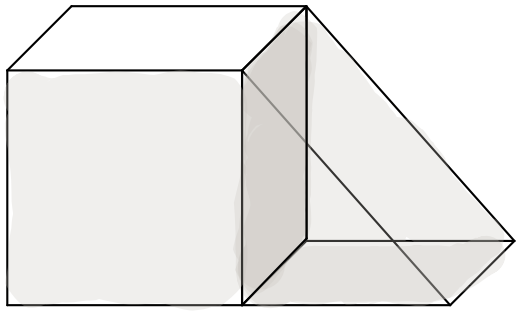
⑥ hybrid systems
- Like a blue ray player
  - Allowar access to Netflix
  - but also reads blue Ray / CD
- Cash registers.

Typical Components
1. User interface
2. Process Management
3. Cpu scheduling
4. Monory Managment

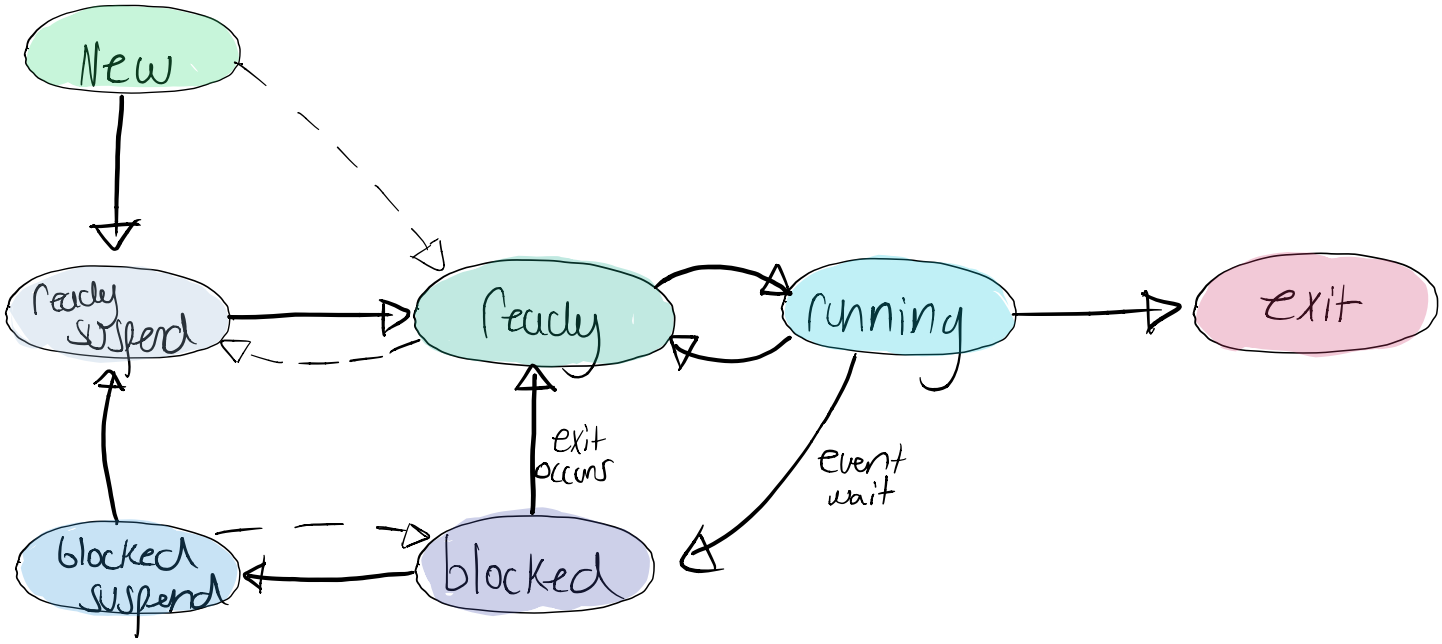
5. Secondary Storage Managment
6. file Managment
7. Interrupt error Managment
8. security features / Monitoring
9. Accounting - tracking information about system.
10. system Managment tools

Process Managment

- Process - running program that use resources.
- State of process
  - running
  - ready
  - blocked - generally waiting on I/O
  - new - loading program
  - Exit - clean up memory.



- Process States



- Suspend States
  1. Swapping
  2. OS reasons
     - process suspected of causing a problem
  3. Timer - periodic processor
  4. User request
  5. Parent Request

- Process description - OS central Data structures
  - Memory tables - allocation of main memory
    - secondary storage
    - protection attributes
  - I/O tables - manage I/O devices
  - file tables - manage files
  - PCB - Process control block
    - Principle data structure used by OS to manage process

PCB contains:

- process name
- process number
- process state
- process context - CPU state
  - used to register contents when it is not running
- Memory use
- Resource use
- process priority
  - used for scheduling
- Relationships

- child/parent, communicating
- accounting information

When we create a process

1. assign a unique process ID
   - for parent process ID = 1
2. Allocate space for process
   - process image
   - memory allocation
3. PCB initialized
4. Linkages set
   - Put in queue
5. All other data structures initalized

Process Switching

- Save context of process
- update PCB
- Move PCB to appropriate queue
- Select another process to execute.
- Update PCB of selected process
- Update memory managment and anything else that

VS

Context Switching

- Interrupt handler returns control to same process

needs updated.

- Restore context