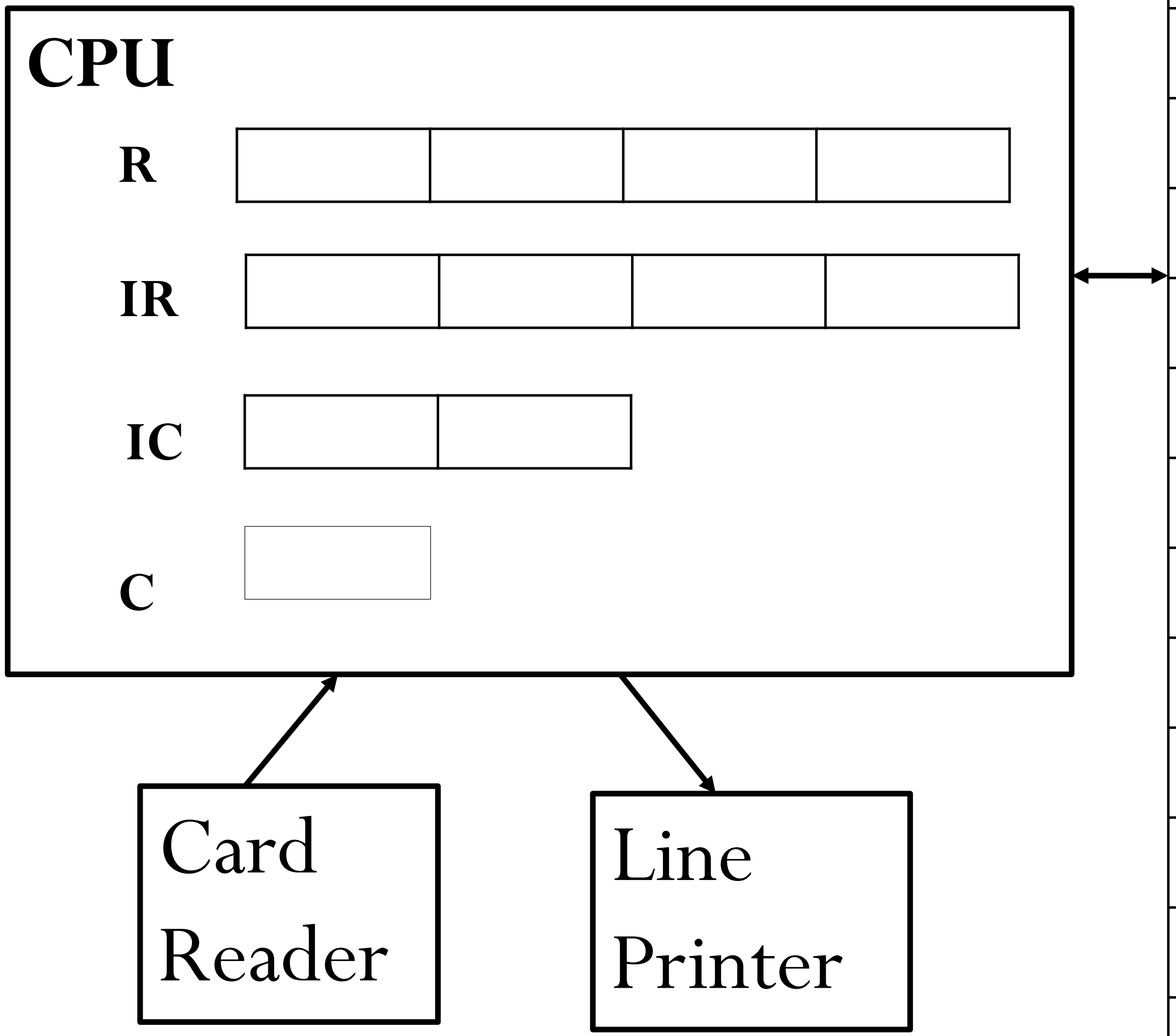
Course Project
Operating System

Operating System-Phase 1

- Virtual Machine
- Instruction Set
- Job
- Program Execution
- Algorithm
- Programming
- Output

Virtual Machine:



Address	Memory Location Content			
00				
01				
02				
03				
:				

Virtual Machine :

1. 1 word =4 Bytes

   Size of total Memory= 100 words=100 x 4 Bytes =400Bytes

2. Memory is divided into Blocks/Pages

   Size of Block (Page)= 10 words =10 x 4 Bytes of each word=40 Bytes.

3. No of pages (Blocks) in memory= Size of Total Memory/ Size of one page

   = 100 words/10 words

   = 10 Pages.

   10 Pages – (0 Page - 9 Page)

4. Card Reader(Input device) read Block of Data /Page -(40 Bytes)

5. Line Printer (Output device) print Block of Data /Page -(40 Bytes)

Virtual Machine Instructions:

SR.No.	Instruction	Operation	Example	Data	Modes
1	GD	Get Data from card reader and placed memory location specified in instruction	GD10,GD20	Block of data (40 Bytes)	Master/Kernel Mode
2	PD	Put (print data) to Line Printer from memory location content specified in instruction.	PD10,PD20	Block of data (40 Bytes)	Master/Kernel Mode
3	LR	Load Register with memory location content specified in instruction	LR12,LR20	Word of data (4 Bytes)	Slave/User Mode
4	SR	Store Register content to memory location specified in instruction.	SR24,SR50	Word of data (4 Bytes)	Slave/User Mode
5	CR	Compare register content with memory location content specified in instruction. Comparison result is equal- C-True Comparison result is not equal-C-False	CR20,CR45	Word of data (4 Bytes)	Slave/User Mode
6	BT	Branch to specified memory location in instruction if toggle register is true.	BT23	Byte data (Toggle Reg.)	Slave/User Mode
7	HLT	Stop execution of program.	HLT	-	Master/Kernel Mode

Virtual Machine Instructions:

Note: GD and PD are I/O instruction. It works on BLOCK of data(40 Bytes)

1. **GD**-Get data from Card Reader and placed specify memory location in instruction.

   e.g. GD10 – Get Data from Card Reader and placed data at specified memory location 10.

2. **PD**- Put (Print) data to Line Printer. Get the data from specified memory location in instruction and put data to Line Printer.

   e.g. PD10-Fetch (Block) of data from memory location 10 and put data to line printer.

Virtual Machine Instructions:

Note: LR and SR (4 Bytes)perform word data transfer operation

3. LR-Load Register content from specified memory location in instruction.

e.g. LR 20- Load data(Word data) into register from memory location 20

LR 23

4. SR-Store Register content to specified memory location in instruction.

e.g. SR 40-Store content of register into memory location 40

SR 45

Virtual Machine Instructions:

Note: Compare instruction works on word data.

5. **CR**-Compare Register content with Memory location content specified in instruction. Result of comparison-Set/Reset (true/false)store in Toggle C register.

   e.g. CR 30 –Compare the content of register with memory location 30

   Result –C-T/F (1/0)

   CR 25

6. **BT**-Branch if true (C Flag status) to specify memory location in instruction

   e.g. BT25 –Branch to memory location 25 if C(toggle register) is true.

7. **H**-HLT –Stop Program Execution.

Virtual Machine Program:

Q. Write an program to display Hello message through Line Printer.

```
$AMJ000100030001
 ; ( Job id /Total Time Limit /Total Line Limit)
GD10PD10H
$DTA
Hello
$END0001
```

Address	Memory Location Content			
00	G	D	1	0
01	P	D	1	0
02	H			
03				
:				
09				
10	H	e	l	l
11	o			
12				
:				
19				
:				
99				

Virtual Machine Program:

Q.Write an program to display following message through Line Printer.

I LIKETHIS PEN OF HIS

Assume Data Card consist following data:

I LIKETHIS PEN OF

Address	Memory Location Content			
00				
01				
02				
03				
:				
09				
10	I		L	I
11	K	E		T
12	H	I	S	
13	P	E	N	
14	O	F		
15				
:				
99				

Virtual Machine Program:

Q. Write an program to display following message through Line Printer.

I LIKETHIS PEN OF HIS

Assume Data Card consist following data:

I LIKETHIS PEN OF

Ans:

```
$AMJ000200050001
GD10LR12SR15PD10H
$DTA
I LIKETHIS PEN OF
$END0002
```

Address	Memory Location Content			
00	G	D	1	0
01	L	R	1	2
02	S	R	1	5
03	P	D	1	0
04	H			
:				
09				
10	I		L	I
11	K	E		T
12	H	I	S	
13	P	E	N	
14	O	F		
15	H	I	S	
:				
99				

Virtual Machine Program:

Q.Write an program to compare data from data card given in the program.

Assume Data Cards consist following data:

ABCD

ABCD

DO NOT

MATCH

Output:

ABCD

ABCD

MATCH

Address	Memory Location Content			
00				
01				
02				
03				
:				
09				
10				
11				
12				
13				
14				
15				
99				

Virtual Machine Program:

Q. Write an program to compare strings given in the program.

Ans.

```
$AMJ000300130003
GD20PD20GD30PD30GD40GD50LR20CR30BT11PD40
HPD50H
$DTA
ABCD
ABCD
DO NOT
MATCH
$END0003
```

Address	Memory Location Content			
00				
01				
02				
03				
:				
09				
10				
11				
12				
13				
14				
15				
99				

Virtual Machine Program:

Q.Write an program to compare data from data card given in the program.

Assume Data Cards consist following data:

VIT

VIIT

IS SAME

NOT SAME

Output:

VIT

VIIT

NOT SAME

Address	Memory Location Content			
00				
01				
02				
03				
:				
09				
10				
11				
12				
13				
14				
15				
99				

Virtual Machine Program:

Q.Write an program to compare two sting VIT and VIIT given in the program.

Ans:

$AMJ000400120003

GD20PD20GD30PD30GD40GD50LR20CR30BT11PD40

HPD50H

$DTA

VIT

VIIT

IS SAME

NOT SAME

$END0004

Address	Memory Location Content			
00				
01				
02				
03				
:				
09				
10				
11				
12				
13				
14				
15				
99				

Algorithm:

Program:

1. Input File
2. Operating System Program
3. Output File

Algorithm:

ASSUMPTIONS:

- Jobs entered without error in input file
- No physical separation between jobs
- Job outputs separated in output file by 2 blank lines
- Program loaded in memory starting at location 00
- No multiprogramming, load and run one program at a time
- SI interrupt for service request

Algorithm:

NOTATION

M: memory; IR: Instruction Register (4 bytes)

IR [0, 1]: Bytes 1, 2 of IR/Operation Code

IR [2, 3]: Bytes 3, 4 of IR/Operand Address

M[&]: Content of memory location &

IC: Instruction Counter Register (2 bytes)

R: General Purpose Register (4 bytes)

C: Toggle (1 byte)

: Loaded/stored/placed into

Algorithm:

```
LOAD
    $m \leftarrow 0$
    While not  e-o-f
        Read next (program or control) card from input file in a buffer
                Control card:    $AMJ, INIT
                                 $DTA, START  EXECUTION
                                 $END, end-while
                Program Card:    Store buffer in memory locations $m$ through $m + 9$
                                 $m \leftarrow m + 10$
    End-While
    STOP
```

Algorithm:

```
INIT
        M: memory- *
        IR: Instruction Register -*
        IC: Instruction Counter Register -*
        R:General Purpose Register -*
        C:Toggle-*
        Buffer-*

START EXECUTION
      IC ← 00
      EXECUTE USER PROGRAM
```

Algorithm:

```
EXECUTE USER PROGRAM (SLAVE MODE)
    Loop
            $IR \leftarrow M [IC]$
            $IC \leftarrow IC+1$
            Examine IR[0,1]
                  LR:     $R \leftarrow M [IR[2,3]]$
                  SR:     $R \rightarrow M [IR[2,3]]$
                  CR:     Compare R and M [IR[2,3]]
                          If equal $C \leftarrow T$ else $C \leftarrow F$
                  BT:     If $C = T$ then $IC \leftarrow IR [2,3]$
                  GD:     $SI = 1$
                  PD:     $SI = 2$
                  H:      $SI = 3$
           End-Examine
    End-Loop
```

Algorithm:

```
MOS (MASTER MODE)
    Case SI of
            1: Read
            2: Write
            3: Terminate
    End case
```

Algorithm:

```
READ
    Read next (data) card from input file in memory locations IR [2,3] through IR [2,3]
    EXECUTE USER PROGRAM

WRITE
    Write one block (10 words of memory) from memory locations IR [2,3] through IR [2,3] + 9  to output file
    EXECUTE USER PROGRAM

TERMINATE
    Write 2 blank lines in output file
    LOAD
```