⟨ Lecture-16 ⟩ 13/11/2019

- Virtual Machine -

- The ~~concept of~~ concept of virtual machine
- Implementation
  - stack operation
  - Memory mapping
- VM code
  - Memory access command
  - Arithmetic and logic command

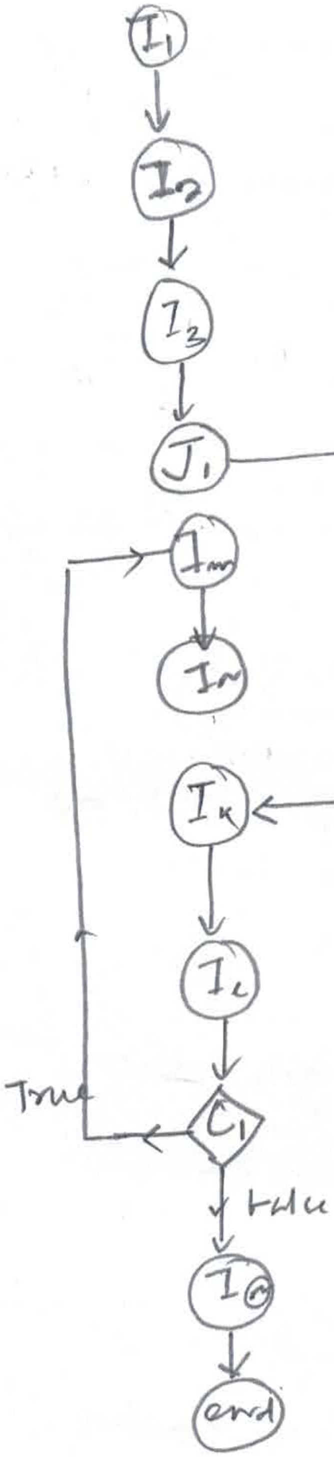
- program flow & function call :-

- Concept of program flow
- Implementation in VM.
  - goto label
  - if-goto label
- Concept of subroutine call & mechanism
- → VM ~~com~~ command
  - function name nn
    - name → name of the function
    - nn → ~~no of parameter to pass~~ no of local varia
  - call name nn
    - nn → no of ~~a~~arguments that have been pusha
  - return ~~Rtt~~

    Return to the calling function.

(2)

Control flow of a program:-



I - an instruction

J - unconditional jump

C - conditional jump

VM command:-

unconditional jump - goto label_name

conditional jump - if-goto labelname

Example:-

```
        push segment index
        push segment index
→ label L1
        pop segment index
```

conditional Jump :-

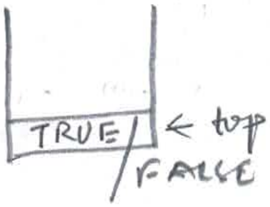
if-goto label-name.

check top of the stack for

0 → false

~~1 → true~~

non. zero → true



- pop the value to appropriate segment.
- execute the Jump instruction

Function call -

- ~~Structure~~ Structure of a function

retuntype function_name parameteres

body of a function

Local variables

local operations

return.

function call and Stack.

Example :- Calling other function.

subroutine main:
call f1
call f2
- - -
- - -
return.

subroutine f1:
call f3
call f4
- - -
return.

subroutine f
c
-
-
return

subrout. f3:
return

(4)

General flow ~~state~~ of function call

① — passing argument. from caller to called.

② — Saving the state of the caller before switching to execute the called

② — Allocating space for the local variables of the called

④ — transfer control to called

⑤ — return values from the called to called

⑥ — Recycling (freeing) the memory space by the called subroutine.

⑦ — Re-instating the state of the called.

⑧ — return the control back to caller