

NAME :- DUBEY KARAN SANJEEV

CLASS :- T.E-4

ROLL NO:- 12, BATCH :- A

3PCC ASSIGNMENT

Q.1. Illustrate the design procedure of Nested Microprocessors with appropriate flow chart.

Q.2. Explain different types of loader.

Q.3. Generate 3 address code in quadruplet form

while ($A < B$) do

If ($C > D$) then

$$X = Y + Z$$

else

$$X = Y - Z$$

Ans 2 There are in total 8 types of loader which are as follows:-

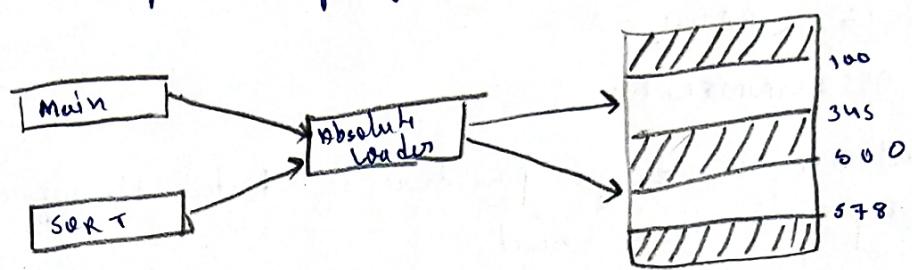
- ① Absolute Loader
- ② Relocating Loader
- ③ Dynamic Loader
- ④ Direct linking Loader.
- ⑤ Assembly / Compile - go Loader.
- ⑥ Boot strap Loader.
- ⑦ Linking Loader.
- ⑧ Relocation Loader.

Out of all this first four are the main loader scheme.

① Absolute Loader - It is a simple type of loading scheme. It accepts the machine language code produced by assembler and place it into main memory.

(i) First - The programmer must specify the address in main memory where the program is to be loaded.

(iii) Second - The programmer must remember the address of multiple sub-programs and their performance.



Absolute loader.

(2) Relocation Loader - It is used to avoid reassembling of all subroutines. When a single subroutine is changed and to perform task of alteration and linking.

Eg - Binary Symbolic Subroutine (BSS), IBM 7294, etc.

(3) Dynamic Loader - Dynamic Loader is the loader that actually intersects the 'calls' and loads the necessary procedure which is called. The drawback is that most of the binding process is postponed until execution time.

(4) Direct Linking Loader - It is the most popular loading scheme and general relocatable loader. Independent translation and inter segment referring is the most advance feature which is used.

(5) Relocation Loader - Relocation is simply moving a program from one area to another in the memory. This is used to add some constant value to each relative address.

Ans 3 solution.

Three address code for a given expression,

which ($n < 0$) do

If ($c > 0$) then

$$x = y + z$$

else

$$x = y - z$$

L1: while ($A < B$) goto L2

goto last

L2: If ($C < 0$) goto L3

goto L4

L3: $t_1 = 4$

$t_2 = 2$

$t_3 = t_1 + t_2$

$x = t_3$

L4: $t_1 = 4$

$t_2 = 2$

$t_3 = t_1 - t_2$

~~x~~ = t_3

last

Quadruple

| op | arg1 | arg2 | result |
|---------|------|-------|--------|
| minus | z | | t_2 |
| + | y | | t_1 |
| = | x | t_3 | |
| minus + | z | | t_2 |
| + | y | | t_1 |
| = | x | t_3 | |

Ans 1 Macro calls within Macros.

- ① Use a stack to create (maintain) order of macro calls.
- ② Create a parameter table at each macro call.
- ③ If a dummy parameter appears in the real parameters field of a called macro, search the parameter table of the calling macro and replace it with the appropriate real parameter.

Find the appropriate macro definition

↳

Build a table for real and dummy parameters

↳

Push LP on to the stack.

↳

LP = first line in the template

↳

Examine the line.

?

Substitute real
parameters

write it to the
output file

LP = LP + 1

evaluate the Boolean
expression

N

True

Y

LP = LP + 1

LP = line no
in the
statement

Find the appropriate
macro definition

Build parameter table

LP = first line in
the template

Pop from stack and
get LP and
current parameters

macro lines

return

MNT - MNT stands for Macro Name Table it consist of all the macro name which are define in the program.