# Unit-4

## 1) Storage orgnanization:

* The compiler demands a block of memory to operating system. for running the compiled code. This block of memory is called run time storage

* The run time storage is divided into different types.

i) The generated target code

ii) Data objects

iii) Information which keeps track of procedure activations

## Generated Code:

* The size of generate code is fixed. hence the target code occupies the statically area of the memory.

* It place target code at the lower end of the memory

## Data objects:

*The amount of memory required by data objects is known as Compiled time

*Data Objects are also have Statically area of memory

## Active Procedures.

*control stack is used to manage the active procedures.

*pascal and C need the run time stack

## 2) Storage allocation strategies:

1) code area

2) Static allocation

3) Stack allocation

4) Heap allocation.

Static allocation:

* In this allocation method the memory is allocated statically at compile time.

* It cannot be changed at run time

* FORTRAN uses the static allocation strategy.

Stack allocation:

* It is also called control stack

* In this strategy, the memory is allocated at run time

* The data structures can be created dyanmically for stack allocation

* It can be changed during run time.

* allocation is slower

## Heap allocation :

* In this storage technique the memory is allocated dynamically

* The heap allocation allocates the continous block of memory when storage is required.

* The free space can be further used by heap manager.

* Linkedlist are mostly used heap management.

## 3) Parameter Passings.

1.) call by Value

2.) Call by Name

3) call by Reference

4) Copy Restore

## 4.) Symbol Table.

**Fixed - length Name.**

| Name | Attribute |
|---|---|
| calculate | |
| sum | |
| a | |
| b | |

Variable length name

| Name | | Attribute |
|---|---|---|
| Starting Index | Length | |
| 0 | 10 | |
| 10 | 4 | |
| 14 | 2 | |
| 16 | 2 | |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | a | l | c | u | l | a | t | e | $ | s | u | m | $ | a | $ | b | $ |

symbol Management tools:

1.) List

2) Hashtables

5) Issues in the design of CDG.

1) Input to the Code Generator

2) Target Program

3) Memory Management

4) Instruction Selection

5) Register allocation

6) choice of Evaluation Order → Order

1) Input to code Generator:

* Input to the code generator is intermediate code generator

1) Linear expression

2) Syntax tree or DAG

3) Three Address code

2) Target program:

i) Absolute ML

ii) Relocatable ML

iii) Assembly Language

*It is output of code generator.

3) Memory Management:

*Runtime memory allocation

*compile time memory allocation

4) Instruction Selection:

Selecting set of instruction

5) Registor allocation

* variables are stored in register allocation

* Register assigments are used for register allocation.