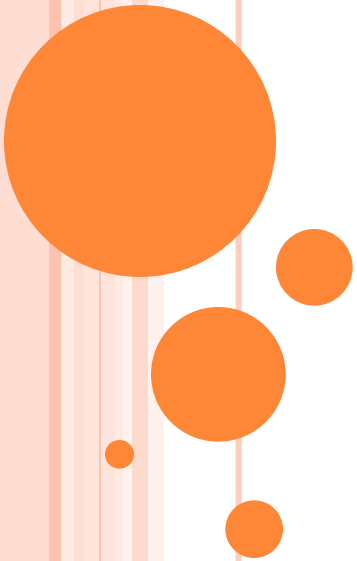
SECTION D

Storage Management

INTRODUCTION

1. Types of Storage Management
2. Programmer memory management
3. System memory management
4. Major elements that requires Storage
5. Phases of storage management



INTRODUCTION

Each and every programming language requires the use of the particular storage technique.

Types of Storage Management

1. **Programmer Control Storage:**

When a management is explicitly done by a user or programmer with the help of certain in-build function like **malloc or free,** that allocate and free the memory as per requirement.

2. **System Controlled Storage Management**

In many programming language the storage management is implicitly done by the system only with the help of various language features

Difference b/w Programmer & system memory management

Programmer Controlled Storage Management	System Controlled Storage Management
In this, programmer explicitly control over the storage with some in build function	In this only system implicitly controlled over the storage using some language features
There is extra burden on the programmer	There is no extra burden on the programmer
It is dangerous to programmer as it may lead to the errors	No such kind of error take place
Some time programmer may conflicts with the controlled system memory mgt.	No such kind of interference of programmer is allowed.
It is easily & efficiently done by programmer as knows the actual requirement of the application	It is quite difficult for the system to determine when to allocate and when to freed

Major elements that requires Storage:

1. Code Segment
2. System Run time Program
3. Storage Management Routine
4. User Define data structure
5. Subprogram Return Points
6. Referencing Environment
7. Temporaries in expression evaluation
8. Temporaries in parameter transmission
9. Input/output buffers
10. Miscellaneous system data

ALONG WITH THIS SOME OPERATION ALSO REQUIRES THE STORAGE LIKE

- Subprogram call and return
- Data structure creation and destruction
- Component insertion and deletion operation

CONTINUED…

Objective of Storage Management:

- To provide the memory space to enable the several processes
- To provide the efficient use of memory
- To protect each program resource

To share the memory space if required

- To make addressing transparent for programmer as much as possible

Features of Storage Management:

1. **Reallocation**
2. **Protection**
3. **Sharing**

CONTINUED…

4.Logical Organization

Program are generally organized in to modules some of them modules may be shared by other some are Read only or some them can be modified .So the storage management will responsible to handle this logical organization.

5. Physical Organization

Memory divided in either **fast RAM or slow secondary memory** so the memory management handles the moving or accessing the information between these two level of the memory.

PHASES OF STORAGE MANAGEMENT

The following the phases that come under the storage management:

1. **INITIAL ALLOCATION**

In the starting of a execution, each piece of storage may be either allocated for some use or free. If it is free then it is available to implement and use it. Any storage management must have some techniques for keep track of free storage as well as allocation of free storage as per need during the execution.

2. RECOVERY

Storage that has been allocated and used and when it will become available it must be recovered by the storage management for reuse. It may performed very simple by means of repositioning stack pointer or complex by means of garbage collection

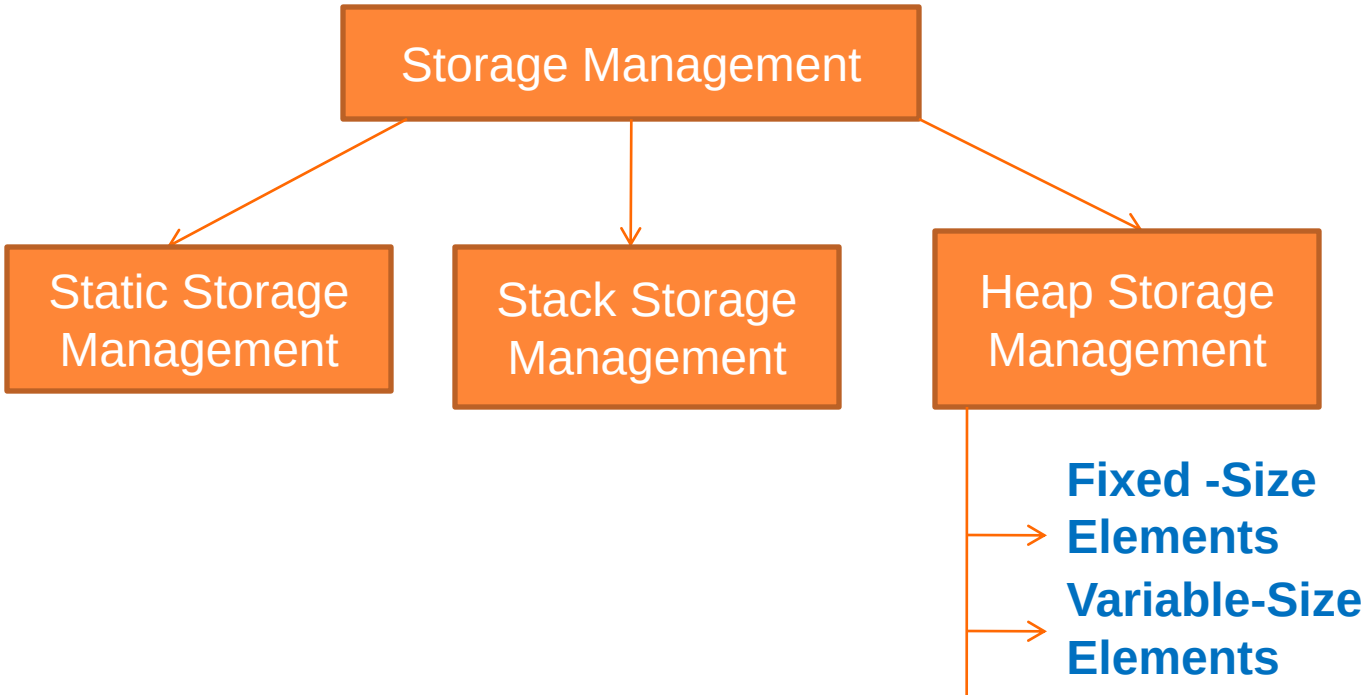
PHASES OF STORAGE MANAGEMENT

3.Compaction and Reuse

storage recovered may be immediately ready for reuse or compaction may be required to construct vary large blocks of the free storage from small pieces.

Storage Management Techniques:

Following are the most basic storage technique:



1. Static Storage Management

This is simplest form of allocation, which is done during translation and remains fixed throughout the execution. Following are the some important point related with the static storage management:

- **Storage for all variables allocated in a static block**
- **Allocation can be done by the translator**
- **could be attached to the code segment**
- **Memory reference can be calculated at the translation time**
- **Subprogram variables use space even if subprogram never called**
- **Recursion not possible**
- **Dynamic data structure are difficult to handle**

1. Static Storage Management

- All storage known at translation time
- Activation records directly associated with code segment
- Procedure call return straight forward

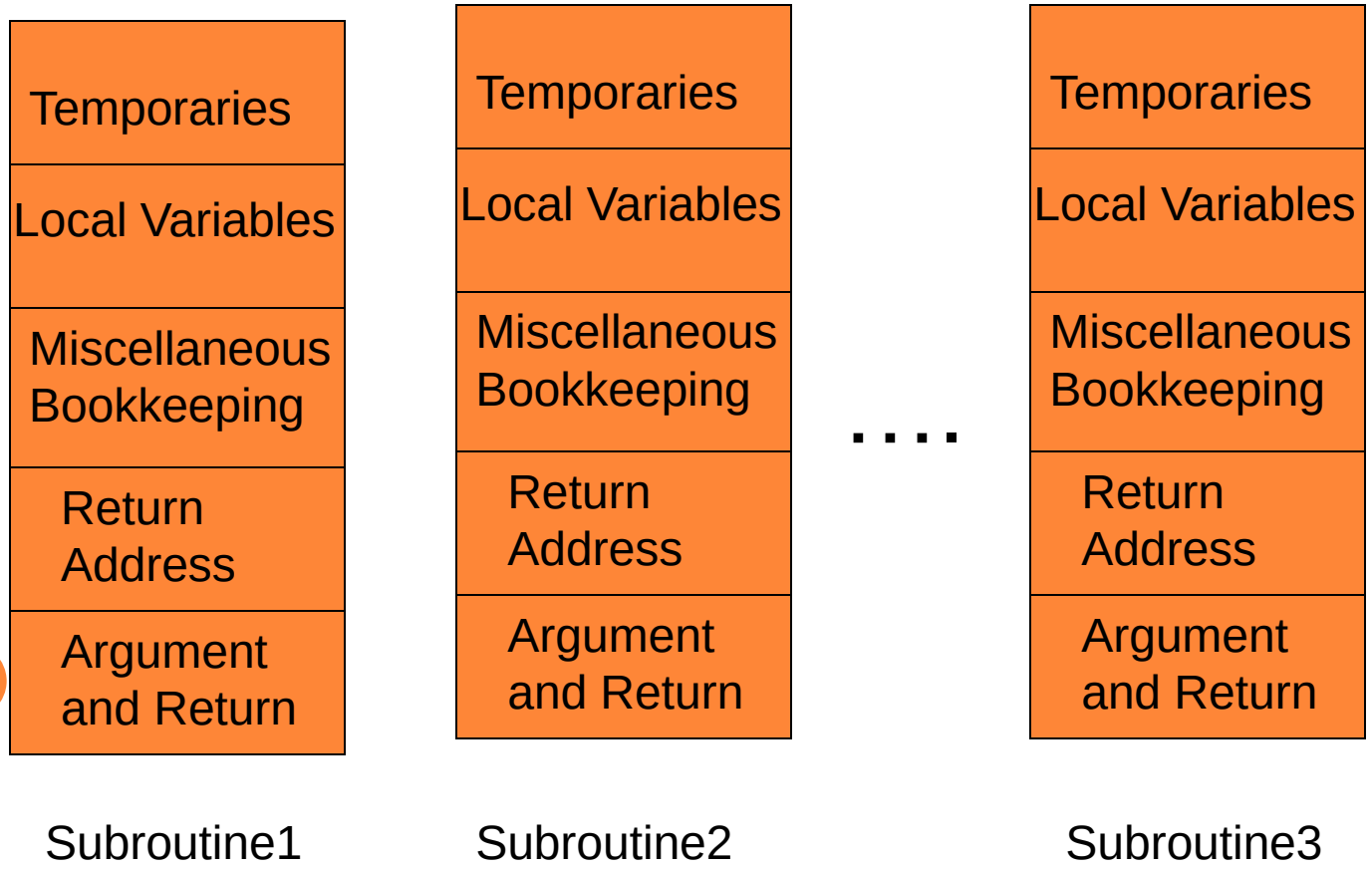
Advantage:

- It is efficient as there is no time or space is expanded for storage management during the execution
- The translator can direct generate l-value address for data items.

Disadvantage:

- It is incompatible with recursive subprograms calls and data structure whose size is dependent on input data

1. Static Storage Management



2.Stack-Based Storage Management

Stack based storage management is based on stack data structure that is **LIFO.** It is needed when language permits the recursion.The following are some important points:

- First allocated- Last freed
- Simple storage recovery ,compaction,and reuse
- In stack based allocation object are allocates in Last-in,first-out

It could be useful in language without recursion as it could save space

- Each subroutine invocation creates a frame or activation record:
  - Arguments
  - Return Address

2.)Stack-Based Storage Management

- Local variables
- Temporaries
- Bookkeeping information

Stack maintained by:

- Calling sequence