

EXPERIMENT - 10

AIM: STUDY OF THE COMPARISON ON VARIOUS LOADER SCHEMES

THEORY:

- What are the functions of Loader?

Ans: The loader performs the following functions:

Allocation: Allocates the space in the memory where the object program would be loaded for Execution. It allocates the space for program in the memory, by calculating the size of the program. This activity is called allocation. In absolute loader allocation is done by the programmer and hence it is the duty of the programmer to ensure that the programs do not get overlap. In reloadable loader allocation is done by the loader hence the assembler must supply the loader the size of the program.

Linking: It links two or more object codes and provides the information needed to allow references between them. It resolves the symbolic references (code/data) between the object modules by assigning all the user subroutine and library subroutine addresses. This activity is called linking. In absolute loader linking is done by the programmer as the programmer is aware about the runtime address of the symbols. In relocatable loader, linking is done by the loader and hence the assembler must supply to the loader, the locations at which the loading is to be done.

Relocation: It modifies the object program by changing the certain instructions so that it can be loaded at different address from location originally specified. There are some address dependent locations in the program, such address constants must be adjusted according to allocated space, such activity done by loader is called relocation. In absolute Loader relocation is done by the assembler as the assembler is aware of the starting address of the program. In relocatable loader, relocation is done by the loader and hence assembler must supply to the loader the location at which relocation is to be done.

Loading: It brings the object program into the memory for execution. Finally, it places all the machine instructions and data of corresponding programs and subroutines into the memory. Thus program now becomes ready for execution, this activity is called loading. In both the loaders (absolute, relocatable) Loading is done by the loader and hence the assembler must supply to the loader the object program.

- Explain Relocation and Linking Concept.

Ans: Relocation is the process of connecting symbolic references with symbolic definitions. For example, when a program calls a function, the associated call instruction must transfer control to the proper destination address at execution. In other words, relocatable files must have information that describes how to modify their section contents, thus allowing executable and shared object files to hold the right information for a process's program image. Relocation entries are these data. Relocation modifies the object program so that it can be loaded at an address different from the location originally specified. Linking, which combines two or more separate object programs and supplies the necessary information.

- Compare Relocating loader and Direct Linking Loader. Explain each one of them in detail.

Ans: Relocating Loader - The relocating loader will load the program anywhere in memory, altering the various addresses as required to ensure correct referencing. The decision as to where in memory the program is placed is done by the Operating System, not the programs header file. This is obviously more efficient, but introduces a slight overhead in terms of a small delay whilst all the relative offsets are calculated. The relocating loader can only relocate code that has been produced by a linker capable of producing relative code. Th

Direct Linking Loader - A direct linking loader is a general relocatable loader and is perhaps the most popular loading scheme presently used. The direct linking loader has the advantage of allowing the programmer multiple procedure segments and multiple data segments and of giving him complete freedom in referencing data. This provides flexible intersegment referencing and accessing ability while at the same time allowing independent translation of

programs. The assembler must give the loader the following information which each procedure or data segment:

1. The length of the segment.
2. A list of symbols in the segment that may be referenced by other segments and their relative location.
3. A list of symbols not defined in the segment but referenced in the segment.
4. Information about where address constants are located in the segment and a description of their revised values.
5. The machine code translation of the source program and the relative address assigned.

- Explain Dynamic Linking Loader in Detail.

Ans: If all the subroutines are loaded simultaneously into the core, then there may be a chance, that the core available may be insufficient for the subroutines, this may lead to further complications. Execution of such program can be possible if all the segments are not required simultaneously to be present in the main memory. In such situations only those segments are resident in the memories that are actually needed at the time execution. But the question arises what will happen if the required segment is not present in the memory? Naturally the execution process will be delayed until the required segment gets loaded in the memory.

Overlay structure is used to specify the interdependency between all the segments. It consists of nodes and edges and segment is represented by the nodes. If the two are on the same path they can lie in the main memory. For solving such problems techniques like segmentation and paging is used. In these the subroutines are loaded into core at a different time because the subroutines in a program are needed at different times e, they may be mutually exclusive, by identifying which subroutine can call other subroutines to produce an Overlay Structure. Overlay Structure is used to:

- Keep in memory only those instructions and data that are needed at any given time
- Overlay structure needed when process is larger than amount of memory allocated to it.
- Implemented by user, no special support needed from operating system. programming design of overlay structures is complex.

The flipper or overlay supervisor is the portion of the loader that actually intercepts the "calls" and loads the necessary procedure.

CONCLUSION:

Hence learnt various loader schemes successfully.