

Assignment - 4

Define loader & what are its function

The loader is program which accept the object program & prepares the program for execution by the computer and initiates the execution.

Function of loader

- ① Allocation:
allocate space in memory for the programs
- ② Linking:
resolve symbolic reference between object & code
- ③ Relocation
Adjust all address dependent location such as address constants to corresponds to the allocated space
- ④ Loading:
physically place the machine instruction and data into memory.

2) Name different types of loading scheme?
Ans different types of loading schemes are

- ① compile and go loader
- ② General loader scheme
- ③ Relocating loaders
- ④ Direct linking loaders
- ⑤ Dynamic loading
- ⑥ Dynamic linking.

3) explain dynamic loading and dynamic binder

Dynamic loading -

If the total amount of core required by all these subroutines exceeds the amount available, as is common with large program on small computers, there is trouble.

There are several hardware techniques, such as paging and segmentation, that attempt to solve this problem!

In this section we will present conventional dynamic loading scheme based upon the use of a binder prior to loading.

Dynamic Binder

- A binder is a program that performs the same function as direct linking loader in binding subroutine together but rather than placing the relocated and linked text directly into memory, its output is the text as a file or card deck.

④ The output file is in a format ready to be loaded and is typically called a load module.

* The loader merely has to physically load the module into core.

* The binder essentially performs the functions of allocation, relocation and linking: the module loader merely perform the function of loading.

*

- There are two major classes of binders
- (a) Core image builders
 - (b) Linkage editor

(4) Explain Four types of cards used in direct linking loader

Ans External Symbol Dictionary cards.

→ The ESD card contains the information necessary to build the external symbol dictionary or symbol table

→ External symbols are the symbols that can be referred beyond the subroutine level

→ There are three types of external symbols

- (a) Segment Definition (SD) - name on START
- (b) Local Definition (LD) - specified on ENTRY card
- (c) External Reference (ER) - specified on EXTERN card

(2) TXT cards

The TXT cards contain block of data and the relative address at which the data is to be placed

→ Once the loader is divided where to load the program it merely adds the program loader address (PLA) to the relative address and moves the data into the resulting location

→ The data on the TXT card may be instructions, non-relocated data or initial values of address constants.

→ The RLD cards

The RLD cards contain the following information:

- (a) The location and length of each address constant that needs to be changed for relocation or linking

- (b) The external symbol by which the address constant should be modified (added or subtracted)
- (c) The operation to be performed (add or subtract)

(3) END cards:

- (a) The end card specifies the end of the object
- (b) If the assembler END card has a symbol in the operand field it specifies a start of execution point for the entire program (all subroutines)
- (c) This address is recorded on the END card.
- (d) This is a final card required to specify the end of a collection of objects
- (e) The 360 loaders usually either a loader terminate (LDT) or END of file (EOF) card

(5) Write specification of database used in pass 1 and pass 2 of direct link loaders.

Ans The database use in pass 1 and pass 2 of direct linking loaders are

- (1) pass 1 data pages
- (1) Input object decks.
- (2) A parameter the initial program load Address (IPLA) supplied by the programme or the operating system that specifies the address to load the first segment.
- (3) A program Load Address (PLA) counter, used to keep track of each segment's assigned location
- (4) A table, the Global External symbol table (GEXT) that is used to store each external symbol and its corresponding core address
- (5) A copy of the input to be used later by pass 2

- ⑥ A printed listing the load map that specifies each external symbol and its assigned value

Pass 2 data base

- ① Copy of object programme inputted to pass 1
- ② The initial program load address parameter (IPLA)
- ③ The program load address counter (PLA)
- ④ The GLEST, prepared by pass 1, containing each external symbol and its corresponding absolute address value.
- ⑤ An array the Local External Symbol Array (LESA) which is used to establish a correspondence between the ESD ID numbers used on ESD and RLD cards, and the corresponding external symbols absolute address value.

