

ASSIGNMENT

Submitted By,

Muhammed Shebin VA

2nd BSc CS

1922k1740

Submitted to,

Bibin Sir

CS Department

Machine-Independent Features

Automatic Library Search

Loading Options

Automatic Library Search

A automatically incorporate routines from subprogram libraries into the program being loaded Standard system library Libraries specified by control statements or by parameters to the loader The subroutines called by the program are automatically fetched from the library, linked with the main program, and loaded. The programmer does not need to take any action beyond mentioning the subroutine names as external references in the source program

Implementation of Automatic Library Search

Linking loader must keep track of external symbols that are referred to, but not defined, in the primary input to the loader. Enter symbols from each Refer record into ESTAB and mark them undefined unless these symbols are already present Make the symbol defined when its definition is encountered At the end of Pass1, the symbols in ESTAB that remain undefined represent unresolved external references. The loader searches the libraries for routines that contain the definitions of these unresolved symbols, and processes the subroutines found by this search process exactly as if they had been part of the primary input stream. Repeat the above library search process until all references are resolved,

because the fetched subroutines may themselves contain external references

Discussions of Automatic Library Search

The presented linking loader allows the programmer to override the standard subroutines by supplying his/her own routines. Directory The loader searches for the subroutines by scanning the Define records for all of the object programs. It can be more efficient by searching a directory giving the name and address of each routine. The same techniques apply equally well to the resolution of external references to data items.

Loader Options

Users can specify options that modify the standard processing of the loader. How to specify options (usually using a special command language): Separate input file Statements embedded in the primary input stream between object programs Loader control statements included in the source program

Loader Options Typical loader options:

Selection of alternative sources of input INCLUDE program-name(library-name)Direct the loader to read the designated object program from a library Deletion of external symbols DELETE csect-nameInstruct the loader to delete the named control sections from the set of programs being loaded Change the external symbols CHANGE name1, name2Cause the external symbol name1 to be changed to name2 wherever it appears in the program Automatic inclusion of library routines LIBRARY MYLIBSpecify alternative libraries to

be searched User-specified libraries are normally searched before the standard system libraries. NOCALL name Specify the external reference name are to remain unresolved.

Example of Using Loader Options

In the COPY program, two subroutines, RDREC and WRREC, are used to read/write records. Suppose that the utility subroutines contain READ and WRITE that perform the same functions of RDREC and WRREC. Without reassembling the program, a sequence of loader commands could be used to use READ and WRITE instead of RDREC and WRREC: INCLUDE READ(UTLIB) INCLUDE WRITE(UTLIB) DELETE RDREC, WRREC CHANGE RDREC, READ CHANGE WRREC, WRITE