

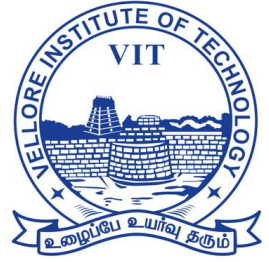


SWE4001 – System Programming

Module 4: Loader and Linkers

Lesson 6 of 6: Loader Design Options

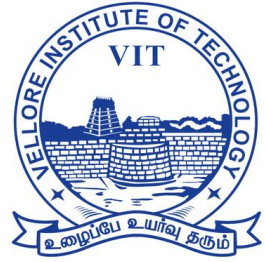
Loader Design Options



- Linkage Editors
- Dynamic Linking

3.4 Loader Design Options

3.4.1 Linkage Editors



❖ Fig 3.13 shows the difference between linking loader and linkage editor.

➤ The source program is first assembled or compiled, producing an OP.

❖ Linking loader

➤ A linking loader performs all linking and relocation operations, including automatic library search if specified, and loads the linked program directly into memory for execution.

❖ The essential difference between a linkage editor and a linking loader

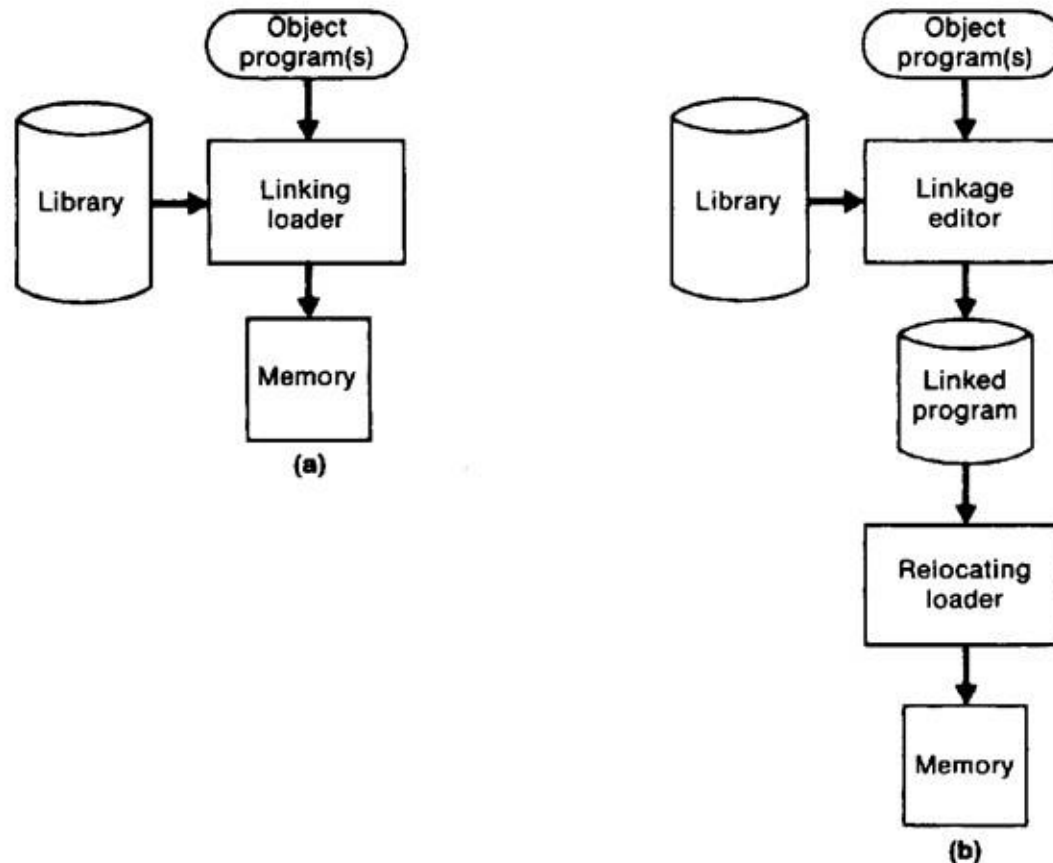
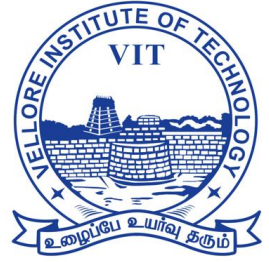


Figure 3.13 Processing of an object program using (a) linking loader and (b) linkage editor.

3.4.1 Linkage Editors



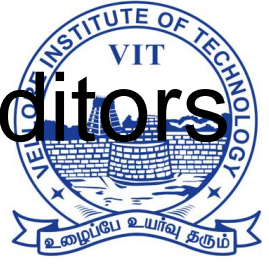
④ Linkage editor

- ⑨ A linkage editor produces a linked version of the program (load module or executable image), which is written to a file or library for later execution.
- ⑨ When the user is ready to run the linked program, a simple relocating loader can be used to load the program into memory.
- ⑨ The only object code modification necessary is the addition of an actual load address to relative values within the program.
- ⑨ The LE performs relocation of all control sections relative to the start of the linked program.

3.4.1 Linkage Editors



- ⑨ All items that need to be modified at load time have values that are relative to the start of the linked program.
- ⑨ If a program is to be executed many times without being reassembled, the use of a LE substantially reduces the overhead required.
- ⑨ LE can perform many useful functions besides simply preparing an OP for execution.



Linking loader vs. linkage editors

- Comparison
 - **Linking Loader**: performs all linking and relocation operations, including library search if specified, and loads the linked program directly into memory for execution
 - **Linkage Editors**: produces a linked version of the program (often called a load module or an executable image), which is written onto a file or library for later execution
 - Resolution of external reference and library searching are only performed once for linkage editor
 - If a program is to be executed many times without being reassembled, the use of a linkage editor substantially reduces the overhead required.
 - If a program is under development or is used infrequently, the use of a linking loader outperforms a linkage editor

3.4.2 Dynamic Linking



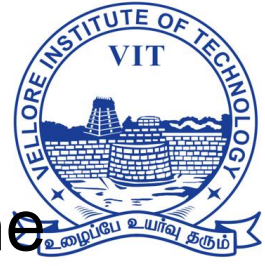
- ④ **Linking loaders** perform these same (linking) operations at **load time**.
- ④ **Linkage editors** perform linking operations **before the program is load** for execution.
- ④ **Dynamic Linking** postpones the linking function **until execution time**.

3.4.2 Dynamic Linking



- ④ Dynamic linking (dynamic loading, load on call)
 - ⑨ Postpones the linking function until execution time.
 - ⑨ A subroutine is loaded and linked to the rest the program when is first loaded.
 - ⑨ Dynamic linking is often used to allow several executing program to share one copy of a subroutine or library.
 - ④ Run-time library (C language), dynamic link library
 - ④ A single copy of the routines in this library could be loaded into the memory of the computer.

3.4.2 Dynamic Linking



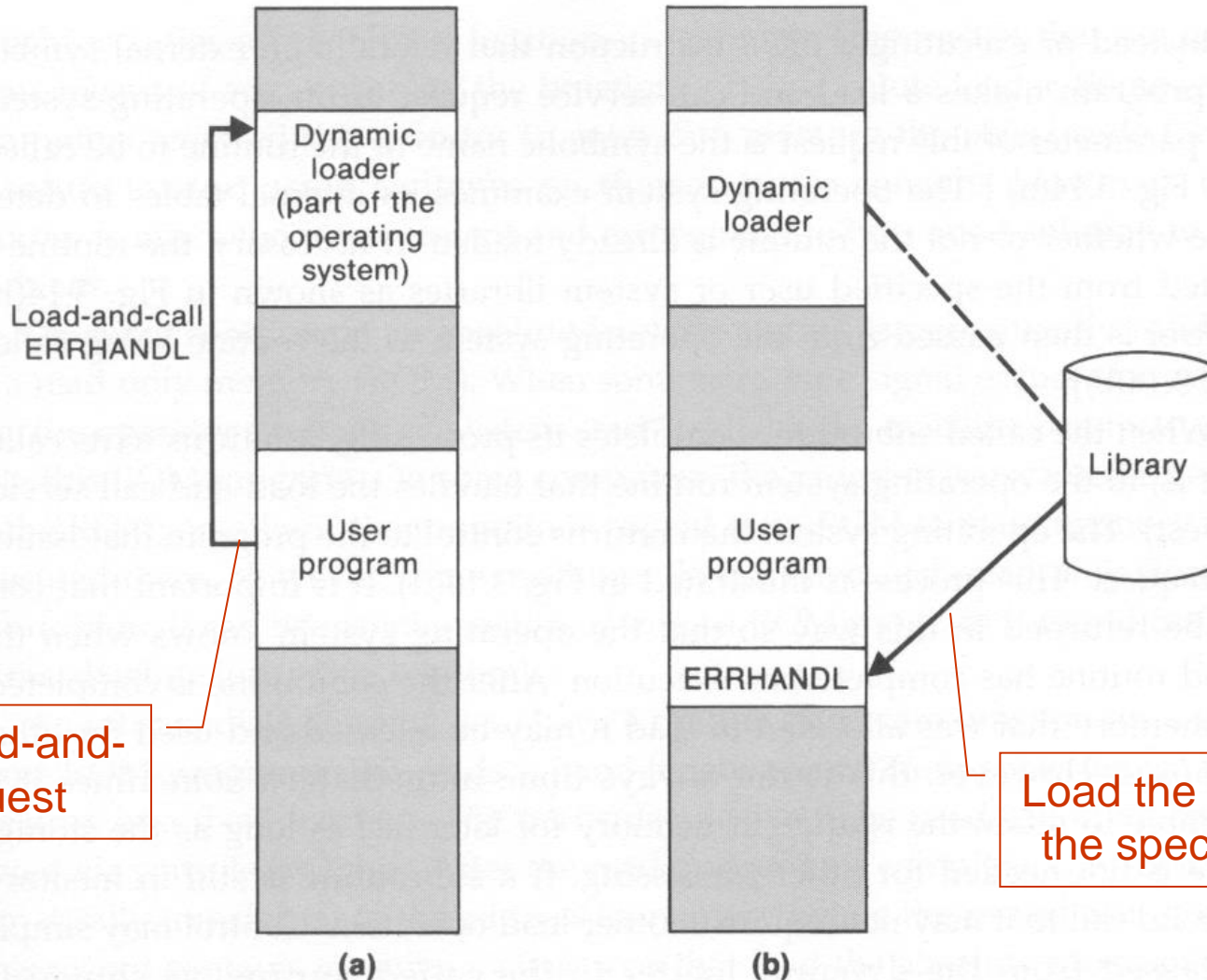
- ④ Dynamic linking provides the ability to load the routines only when (and if) they are needed.
 - ⑨ For example, that a program contains subroutines that correct or clearly diagnose error in the input data during execution.
 - ⑨ If such error are rare, the correction and diagnostic routines may not be used at all during most execution of the program.
 - ⑨ However, if the program were completely linked before execution, these subroutines need to be loaded and linked every time.

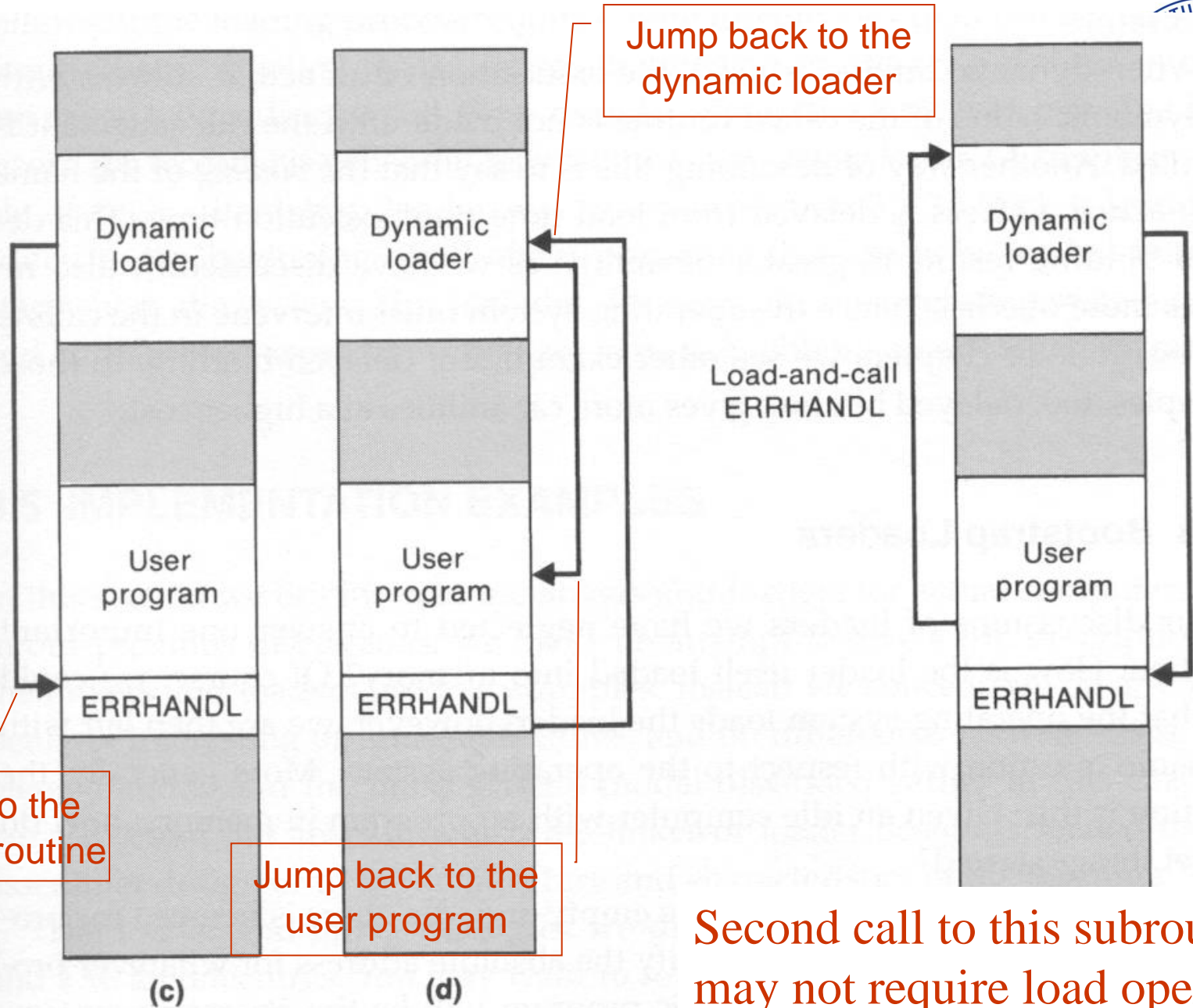
3.4.2 Dynamic Linking



- ④ Dynamic linking avoids the necessity of loading the entire library for each execution.
- ④ Fig. 3.14 illustrates a method in which routines that are to be dynamically loaded must be called via an operating system (OS) service request.

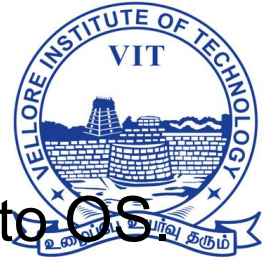
Example of Dynamic Linking





Second call to this subroutine may not require load operation.

3.4.2 Dynamic Linking



- ④ The program makes a load-and-call service request to OS. The parameter argument (ERRHANDL) of this request is the symbolic name of the routine to be loaded.
- ④ OS examines its internal tables to determine **whether or not the routine is already loaded**. If necessary, the routine is loaded from the specified user or system libraries.
- ④ Control is then passed from OS to the routine being called.
- ④ When the called subroutine completes its processing, OS then returns control to the program that issued the request.
- ④ If a subroutine is still in memory, a second call to it may not require another load operation.