Minor Project- Report

Aug-2019-2020

Course Faculty: Dr Vindhya M Course Name & code: SYSTEM SOFTWARE

Semester:6 Date:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TITLE OF THE PROJECT | LOADER IMPLEMENTATION | | | |
|  |  | | | |
| STUDENT NAME | Siddharth Koul | Shreyas M | Shreyans Sonthalia | SHYALE SHARAN |
| USN | 1DS17CS112 | 1DS17CS110 | 1DS17CS109 | 1DS17CS111 |
| PROJECT ABSTRACT : | In computer systems a loader is the part of an operating system that is responsible for loading programs and libraries. It is one of the essential stages in the process of starting a program, as it places programs into memory and prepares them for execution. Loading a program involves reading the contents of the executable containing the program instructions into memory, and then carrying out other required preparatory tasks to prepare the executable for running. Once loading is complete, the operating system starts the program by passing control to the loaded program code.  All operating systems that support program loading have loaders, apart from highly specialized computer systems that only have a fixed set of specialized programs. Embedded systems typically do not have loaders, and instead, the code executes directly from ROM. In order to load the operating system itself, as part of booting, a specialized boot loader is used. In many operating systems, the loader resides permanently in memory, though some operating systems that support virtual memory may allow the loader to be located in a region of memory that is pageable.  In the case of operating systems that support virtual memory, the loader may not actually copy the contents of executable files into memory, but rather may simply declare to the virtual memory subsystem that there is a mapping between a region of memory allocated to contain the running program's code and the contents of the associated executable file.  In this project , we will be implementing an absolute loader. | | | |

|  |  |
| --- | --- |
| Introduction | An absolute loader is the simplest type of loader. The operation of absolute loader is very simple. The object code is loaded to specified locations in the memory. At the end the loader jumps to the specified address to begin execution of the loaded program. The advantage of absolute loader is simple and efficient. But the disadvantages are, the need for programmer to specify the actual address, and, difficult to use subroutine libraries. |
|  |  |
| Design &  Algorithm | **Object**  **Program**  **Absolute**  **Loader**  **Object**  **program**  **ready for**  **execution**  **Memory**  1000  2000  **Begin** read Header record verify program name and length  read first Text record **while** record type is <> ‘E’ **do begin**  {if object code is in character form, convert into internal representation}  move object code to specified location in memory read next object program record **end**  jump to address specified in End record **end** |
|  |  |
| Project Source Code Link (Github/ Google DRive) |  |
|  |  |
| Conclusion /FUTURE ENHANCEMENT |  |