

ECT-206

ASSIGNMENT

Jibin Babu Amakkattu
St ECE-B

22

(1) Explain about Assembler, Interpreter, Compiler, Linker, Loader, Debugger.

Assembler :

- Assembler translates assembly language programme into machine code.
- output of Assembler is called 'object file', which contains a combination of machine instruction as well as data required to place these instructions in memory.

There are two types of Assemblers:

1. One pass Assembler: It assigns the memory addresses to the variables and translates the source code into machine code in the first pass simultaneously.
2. Two pass Assembler: It reads the source code twice. In the first pass it reads all variables and assigns them memory addresses. In the second pass, it reads the source code and translates the code into object code.

Interpreter:

- Computer program that translates statements of a program into machine code.
- translates only one statement at a time.
- It reads one statement, translates it and execute it and then move to the next line.
- If an error occurs, it stops execution and reports it.
- Machine codes produced by interpreter are not saved.
- Occupies less memory space, so can be used in smaller systems with limited memory space.

Compiler :

- Translate high level language program into a target language.
- target language can be machine language or assembly language, by default it is machine language (object code).
- More intelligent than assembler
- checks all kinds of limits, range, errors etc.
- Program run time is more, occupies longer memory.
- slow speed.
- goes through the entire program and then translates it.

self-compiler: If a compiler runs on computer and produce machine code for the same computer, then it is known as self compiler.

Cross compiler: Runs on a computer and produce machine code for another computer.

Linker:

- In high level languages, some built in header files or libraries are stored and they are predefined.
- Those libraries contain some basic functions which are essential for executing the program. These function are linked to the libraries by a program called linker.
- If linker does not find a library of a function, it informs to compiler and it generates an error.
- Compiler automatically invokes the linker as the last step in compiling a program.
- Longer programs are divided into smaller subprograms called modules and they must be combined to execute the program. This combining process is done by the linker.
- Linker is a computer program that merges various object files together in order to make an executable file.
- Major task is to locate and search referenced module/routine in a program, to determine the memory location where the codes will be loaded.

Loader:

- It is a program that loads machine code of a program into the system memory.
- Essential stage in starting a program, because it places programs into memory and prepares them for execution.
- Involves reading the contents of executable file into memory.
- Mostly the loader is permanently resident in memory.
- Responsible for loading executable file into memory and execute them.
- Calculates the size of a program and creates memory space for it.
- Initializes various registers to initiate execution.

Debugger:

- Software program used to test and find bugs in programs.
- Assists the detection and correction of errors in other computer programs.
- Allows to view another program line by line.
- Other features are running a program step by step, stopping the program to see its current state, tracking the variable values etc.
- Some debugger programs can also modify a program while it is running.

(2) Explain how a HLL program is executed as machine language in a processor

1. High level code: The high level code, written serves as an input for the execution process
2. Compiler: Translates the high level code into machine code.
3. Assembler: Translates assembly code into machine code that the processor can execute.
4. Linker: Combines object files generated by the compiler or assembler. It creates single executable file ready for execution.
5. Loader: Loads the executable file into memory, allocating the necessary resources for the program's execution.
6. Interpreter: Executes the high level code directly translating and executing it line by line during runtime.
7. Memory access: Processor reads/writes variable and data structure from memory
8. Debugger: Assists in monitoring and controlling the execution of the program.

In summary, the compiler and assembler translate HLL and assembly code into machine code. Linker combines the generated code and resolves dependencies. Loader loads the code into memory and the interpreter executes it directly or processor executes the machine code. Throughout the process debugger helps in monitoring and analyzing the programs behavior.