Name - MD. Rakib Hasan.

ID - 2216080021.

Subject code / Name - Compiler.

Batch - 08 (EV).

Ans. to the Q. No-01

(a)

Types of Compiler :-

There are various types of compailers which are as follows:-

Traditional Compilers (C, C++, and Pascal) :-

These compilers transform a source program in an HLL into its similar in native machine program or object program.

Interpreters (LISP, SNOBOL, and Java 1.0) :-

These compilers first convert source code into intermediate code, and then interprets it to equivalent matchine code.

Cross Compilers :-

These are the compilers that run on one machine and make code for another machine,

12/17/2023 23:41

Incremental Compailers :-

Incremental compailer is a compailer, which executes that recompailation of only a changed source instead of compiling the complete source code.

Converters (e.g. COBOL to C++) :-

These programs will be compiling from one high-level language to another.

Just-In-Time Compilers (Java, Microsoft.NET) :-

These are the run time compilers from intermediate language (byte code, MSIL) to executable code or native machine code.

Single Pass Compiler :-

In a single pass compiler, when a line source is processed it is scanned and the tokens are extracted.

Multi-pass compiler :-

The compiler scans the input source once and makes the first modified structure, therefore scans the first-produced form and makes a second modified structure

Ahead-of-Time (AOT) Compilers :-

These are the pre-compilers to the native code

12/17/2023 23:41

for Java and .NET.

Binary Compilation -

These compilation will be compiling the object code of one platform into the object code of another platform.

Phases of a Compiler :-

A compiler is a software program that converts the high-level source code written in a programming language into low-level machine code that can be executed by the computer hardware.
The typical phases of a compiler are :-

1. Lexical Analysis :-

The first phase of a compiler is lexical analysis, also known as scanning. This phase reads the source code and breaks it into a stream of tokens, which are the basic units of the programming language.

2. Syntax Analysis :-

The second phase of a compiler is syntax analysis, also known as parsing. The output of this phase is usually an Abstract Syntax Tree (AST).

12/17/2023 23:41

3. Semantic Analysis :-

The third phase of a compiler is semantic analysis. This phase is checked wheather the code is semantically correct. The compiler also checks for another semantic errors, such as undeclared variables and incorrect function calls.

4. Intermediate Code Genaration :-

The forth phase of a compiler is intermediate code genaration. This phase genarates an intermediate representation of the source code to ~~improve the performance~~ that can be easily translate into matchine code.

5. Optimization :-

The fifth phase of a compiler is optimization. This phase are applies various optimization techniques to the intermediate code to improve the performance of the generated matchine code.

6. Code Genaration :-

The final phase of a compiler is code generation. This phase takes the optimized intermediate code and genaraters the actual matchine code that can be executed by the target hardware.

12/17/2023 23:41

to be installed in their system, for executing the shared executable file of your source code.

Disadvantages of Compiler :-

There are a few potential disadvantages of using a compiler in software devlopment :

1. Compilation time :-
Depending on the size and complexity of the source code, compilation can take a significant amount of time.

2. Error Detection :-
Compilers scan only detect syntax errors and certain semantic errors and may not catch all errors in the source code.

3. Portability :-
Programs compiled from high-level languages may not be able to run on other platforms or architectures without being compileed.

4. Execution Speed :-
Programs Compiled from high level languages may not be as fast as programs written in low-level languages, as the compiled code may include additional instructions for the compiler to interpreet.

5. Lack of flexibility :-
Compilers can limit the flexibility of programs since changes often requires recompilation.

12/17/2023 23:42

(b) Advantages of compiler :-

There are several advantages to using a compiler :-

1. <u>Improve Performance :-</u>
compiled code tends to run faster than interpreted code because it has been translated into matchine code that can be directly executed by the computers processore .

2. <u>Portability :-</u>
Compilers allow programmers to write code in a high-level programming language that can be easily translated into matchin code for a variety of diffrent platforms .

3. <u>Increased Security :-</u>
compilers can help improve the security of software by performing a number of checks on the source code , such as checking for syntax errors and enforcing type safety .

4. <u>Debugging Support :-</u>
Most compilers include a number of debugging tools that can help programmers find and fix errors in their code .

5. <u>No dependencies :-</u>
Your client or anyone else doesn't need any compiler , interpeter or third party program

12/17/2023 23:42

(c) Why to learn compiler design:-

Compiler design helps full implementation of High-Level programming languages. Support optimization for Computer Architecture Parallelism. A compiler is a program that translates a high-level language into a low-level language.

Top-Down Parsing:-

Top Down parsing searches for a production rule to be used to construct a string.
Top-down approach starts evaluating the parse tree from the top and move downwards for parsing other nodes.
Top-down parsing attempts to find the left most derivation for a given string.
Top-down parsing uses leftmost derivation.

Bottom-up Parsing:-

Bottom-up Parsing searches for a production rule to be used to reduce a string to get a starting symbol of grammar.
Bottom-up parsing attempts to reduce the input string to first symbol of the grammar.
Bottom-up parsing uses the rightmost derivation.

12/17/2023 23:42

Ans. to the Q. No – 02

(a) Language processing System :-

The computer is an intelligent combination of software and hardware.
The hardware consider considers instructions as electronic charge, which is equivalent to the binary language in software programming. The binary language has only 0s and 1s. To enlightn, the hardware code has to be written in binary format, which is just a series of 0s and 1s.
This is known as a Language processing system.

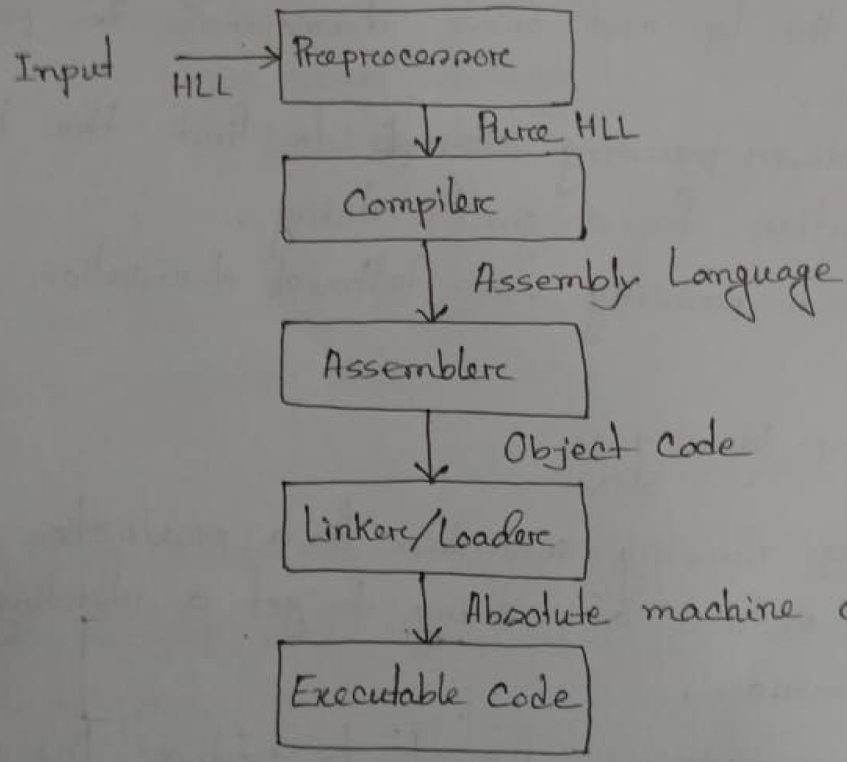


Fig: Language Processing System

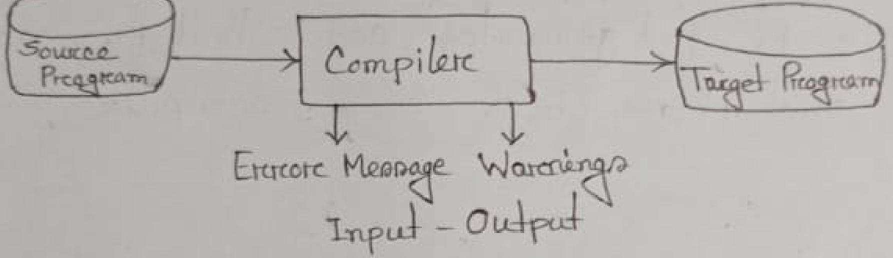
12/17/2023 23:42

Preprocessor :-

It includes all header files and also evaluates whether a macro is a piece of code that is given a name. The purpose of macros is either to automate the frequency used for sequences or to enable more powerful abstraction is included. It takes source code as input and produces modified sources code as output.

Compiler :-

The compiler takes the modified code as input and produces the target code as output.



Assembler :-

The assembler takes the target code as input and produces real locatable machine code as output.

Linker :-

Linker or link editor is a program that takes a collection of objects and combines them into an executable program.

Loader :-

The loader keeps the linked program in the main memory.

Executable Code :-

It is low-level and matchin-specific code that the machin can easily understand. Once the job of the linker and loader is done the object code is finally converted it into executable code.

(b) Cross Compiler :-

A cross compiler is a compiler capable of creating executable code for a platform other than the one on which the compiler is running. For example, a compiler that runs on a PC but generates code that runs on an Android smartphone is a cross compiler.

(c) Source Compiler :-

A source to source compiler (S2S compiler) is also referred to by three name, the first is the source to source translator, the second is transcompiler and the third one is transpiler.

A S2S compiler is given as input the source code of a program to which it returns a source code with the overall same functionality in the same or diffrent programming language.

12/17/2023 23:43