29/01/22

Tutorial-03

U19CS009

Page No.:
Date : / /

Q-1 Describe in detail about any 4 system softwares.

Ans a) **Operating System** :- It manages computer hardware and software resources. It provides common services for computing. It acts as resource manager.
Following are it's functions:-
1. interface b/w user and hardware through GUI
2. memory management
3. Processor management
4. file management and many more
5. Security

b) **Device Drivers** :- This controls specific hardware device that enables different hardware devices of communication with computer OS. Without drivers the computer could not send and receive data correctly to hardware devices.

c. **Utility software** :- This software analyses and maintain a computer. This are b/w system sof and appliration software. eg. anti-virus, file management tools.

d. **Language processor** :- This software converts source code into machine code. i.e, a higher level language into lower level language, comprehable to machine. This includes compilers, interpreters, assemblers etc.

Q-2 Define assembler, compiler and interpreter

a) **Compiler** => This is a system software that reads the complete source code in one go written in a high level language and translates it into an equivalent

program in machine language. Source is converted to object code if it is error free, or else it point out where may be the error.

b. Assembler :- It is a system software that converts a assembly level language into machine code, which is binary code. They create instructions for each line in assembly code.

c) Interpreter :- It translates single statement of source code into machine code and executes the statement before moving to next statement. If an error is founds it stops there and sends error message of that line. The execution is done without converting them to an object code. eg. Python, Matlab etc

Q-3 What is system software? Give some applications of Operating systems.

Ans System softwares are special softwares that provides base to other application software. Some eg. are OS, device drivers, compilers etc. These softwares are programmed such that they interact with hardware and application software at a very low level. They serves as interface b/w hardware and application software.

-> Following are applications of OS :-

1.) Buffering
2.) SPOOLING
3.) Memory management
4.) Partitioning
5.) Virtual memory
6.) File management

Page No.:
Date: / /

Q-4 What are two parts of compilation? Explain the various phases of compilation.

Ans The two main parts of Compilation are :-

a) Analysis -> This is also known as front-end of the compiler. In it the compiler reads the source program, divides it into core parts and then checks for lexical, grammar and syntax errors. The analysis phase generates an intermediate representation of the source code, program table, which should be fed to synthesis phase as input.

b) Synthesis -> This is also known as back end of compiler. It generates the target program with the help of intermediate code representation and symbol table.

=> Phases of Compilers

a.i) Lexical Analyzer :- It is called scanner. It takes output of pre-processor as input and this is purely in high level language. It reads characters from source code program and groups them into lexeme. It also removes lexical errors (characters etc), comment and white spaces.

a.ii) Syntax Analyzer :- It is also called parser. It constructs the parse tree. It takes all tokens generated by lexical analyzer and uses CFGs to build tree. Parse tree / Derivation tree is used to check ambiguity in given grammar. Syntax error can be detected at this level.

b iii) Semantic Analyzer :- It verifies the parse tree, whether it's meaningful or not. It furthermore produces a verified parse tree. It also does type checking, Label checking and flow control checking.

b. iv) Intermediate code Generator :- It generates a code, which can be readily executed by a machine. Till intermediate code, everything is same for all compilers, after which, next two phases convert them in machile language, which are platform dependent.

b. v) Code optimizer :- It transforms the code so that it consumes fewer resources and produces more speed.

b. vi) Target Code generator :- It writes a code that the machine can understand and also register allocation, instruction selection etc. The output is dependent on type of assembler. The output is such that it can be relocated. This is the final stage of compilation.

Q-5 D/B tokens, patterns and lexeme

a) Tokens : meaningful sequence of characters.

b) Patterns :- Rule associated with token which describe a set of string which is produced as output. This rules are called pattern.

c) Lexeme :- A sequence of characters in source program that is matched by pattern for a token.