

www.csepedia.com

Punjab Technical University

B.Tech. – Computer Science & Engineering (Sem. – 4th)

System Programming

Subject Code: CS-210

2 Marks Questions:-

Q:-1) What is Syntax Analysis? (May 2012)

Ans:-1) **Syntax Analysis**:- It receives valid tokens from Scanner (Lexical Analysis Phase) and checks it's against Grammar and produces valid syntactical constructs. This process is also known as Parsing.

Q:-2) What is Lexical Analysis. (May 2012)

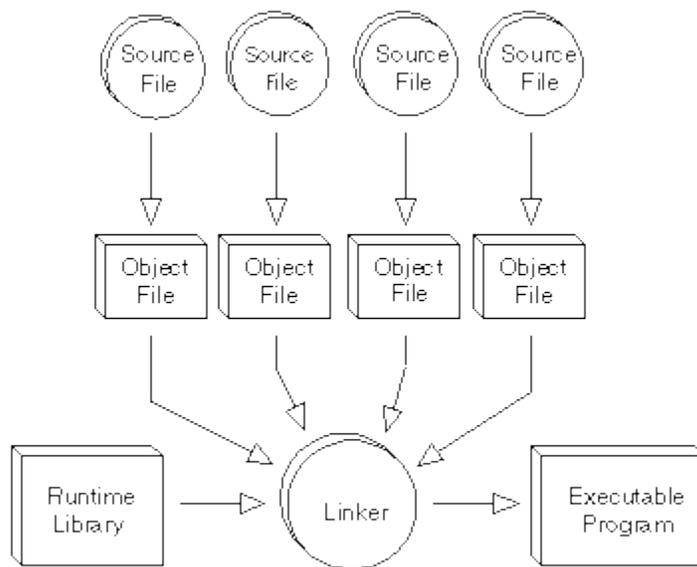
Ans:-2) Lexical Analysis is basically a tool for generating Scanner. So Lexical only deals with creation of Uniform syntax for Literals and Identifiers tables. The Lexical Analyzer breaks up input streams to elements known as Tokens.

Q:-3) What is Debugging? (May 2012)

Ans:-3) **Debugging** is the process of locating and fixing or bypassing bugs (errors) in computer program code or the engineering of a hardware device.

Q:-4) What is Linker? (May 2012)

Ans:-4) **Linker** is a program that combines object modules to form an executable program. Many programming languages allow you to write different pieces of code, called *modules*, separately. This simplifies the programming task because you can break a large program into small, more manageable pieces. Eventually, though, you need to put all the modules together. This is the job of the linker.



Q:-5) What is Loader? (May 2012)

Ans:-5) **Loader** is an operating system utility that copies programs from a storage device to main memory, where they can be executed. In addition to copying a program into main memory, the loader can also replace virtual addresses with physical addresses.

Q:-6) What is Editor? (May 2012)

Ans:-6) **Editor** Sometimes called *text editor*, a program that enables you to create and edit text files. There are many different types of editors, but they all fall into two general categories:

- **Line editors:** A primitive form of editor that requires you to specify a specific line of text before you can make changes to it.
- **Screen-oriented editors:** Also called *full-screen editors*, these editors enable you to modify any text that appears on the display screen by moving the cursor to the desired location.

The distinction between **editors** and **word processors** is not clear-cut, but in general, word processors provide many more formatting features. Nowadays, the term editor usually refers to source code editors that include many special features for writing and editing source code.

Q:-7) What is Finite State Machines? (May 2012)

Ans:-7) A **Finite-State Machine (FSM)** or **finite-state automaton** or simply a **state machine**, is a mathematical model of computation used to

design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of *states*. The machine is in only one state at a time; the state it is in at any given time is called the *current state*. It can change from one state to another when initiated by a triggering event or condition; this is called a *transition*. A particular FSM is defined by a list of its states, and the triggering condition for each transition.

Q:-8) What is Kernel? (May 2012)

Ans:-8) Kernel: - The Central module of an operating system. It is the part of the operating system that loads first, and it remains in main memory. Because it stays in memory, the kernel is responsible for memory management, process and task management, and disk management.

Q:-9) Define Grammar of Language? (May 2012)

Ans:-9) In formal language theory, a **Grammar** is a set of formation rules for strings in a formal language. The rules describe how to form strings from the language's alphabet that are valid according to the language's syntax. A grammar does not describe the meaning of the strings or what can be done with them in whatever context—only their form.

Q:-10) What is Intermediate Code? (May 2012)

Ans:-10) Intermediate codes are machine **independent** codes, but they are close to the Machine Instructions.

The given program in a source language is converted to an equivalent program in an intermediate language by the intermediate code generator. Intermediate language can be many different languages, and the designer of the compiler decides this intermediate language.

- **Syntax Trees** can be used as an intermediate language.
- **Postfix notation** can be used as an intermediate language.
- **Three-Address code** can be used as an intermediate language

Q:-11) What is Vi Editor? How it is useful to us? (May 2010)

Ans:-11) The Vi editor (short for Visual Editor) is a screen editor which is available on almost all Unix systems. Once you have learned vi, you will find that it is a fast and powerful editor. vi has no menus but instead uses combinations of keystrokes in order to accomplish commands.

Vi has two modes: the command mode and the insert mode. It is essential that you know which mode you are in at any given point in time. When you are in command mode, letters of the keyboard will be interpreted as commands. When you are in insert mode the same letters of the keyboard will type or edit text. vi always starts out in command mode.

Q:-12) Define Instruction Counter? (May 2010)

Ans:-12) Instruction Counter: - A counter that indicates the location of the next computer instruction to be interpreted. Also known as location counter; program counter; sequence counter.

Q:-13) Define Static Bindings. (May 2010)

Ans:-13) Static binding means reference are resolved in compile time. The choice of which function to call is made at compile time.

Q:-14) Give the difference b/w Multiprogramming and Multiprocessing. (Dec 2012)

Ans:-14) Multiprogramming

- Multiprogramming is basically a type of parallel processing that can be used in many different environments.
- In Multiprogramming, a number of processes run on single processor.

Multiprocessing

- Multiprocessing is a method of computing in which different parts of a task are distributed between two or more similar central processing units.
- In Multiprocessing, a process is run on multiple processors.

Q:-15) What is the difference b/w a Macro and a Subroutine. (Dec 2012)

Ans:-15) 1. A **Macro call** is an instruction to replace the macro name with its body, whereas **Subroutine call** is an instruction to transfer the program's control to the subroutine's definition with all parameters, if required.

2. A macro call results in macro expansion, whereas subroutine call results in execution.

3. Macro expansion increases the size of the program but subroutine execution doesn't affect the size of the program.

4. Macro expansion doesn't affect the execution speed of the program much in comparison to subroutines affect the execution speed of the program

Q:-16) What is System Programming. (Dec 2012)

Ans:-16) System programming is the activity of computer programming system software. The primary distinguishing characteristic of systems programming when compared to application programming is that application programming aims to produce software which provides services to the user (e.g. word processor), whereas systems programming aims to produce software which provides services to the computer hardware (e.g. disk defragmenter). It requires a greater degree of hardware awareness.

Q:-17) What is the use of LORG pseudo-op? (Dec 2012)

Ans:-17) LORG: Pseudo-op tells the assembler to place the encountered literals at an earlier location. (Used in the case where we have very long program).

Q:-18) Give the difference between BALR and USING. (Dec 2012)

Ans:-18) BALR is one machine mechanism for loading the base register, on the other hand

USING tell the assembler which register will be used as a base & what it contains.

Q:-19) What is Relocation? Why it is needed? (Dec 2012)

Ans:-19) Relocation is the process of assigning load addresses to various parts of a program and adjusting the code and data in the program to reflect the assigned addresses.

A linker usually performs relocation in conjunction with **symbol resolution**, the process of searching files and libraries to replace symbolic references or names of libraries with actual usable addresses in memory before running a program.

Q:-20) What is Bootstrap Loader? (Dec 2012)

Ans:-20) Bootstrap Loader is a program that resides in the computers ROM, or other non-volatile memory that automatically executed by the

processor when turning on the computer. The bootstrap loader reads the hard drives boot sector to continue the process of loading the computers operating system.

Q:-21) What is the difference between DFA and NFA? (Dec 2012)

Ans:-21) (1)

- For Every symbol of the alphabet, there is only one state transition in DFA.
- We do not need to specify how the NFA reacts according to some symbol.

(2)

- DFA can be understood as one machine.
- NFA can be understood as multiple little machines computing at the same time.

(3)

- DFA will reject the string if it ends at other than accepting state.
- If all of the branches of NFA dies or rejects the string, we can say that NFA reject the string.

Q:-22) What is the use of Symbol Table. (Dec. 2011)

Ans:-22) Symbol Table is the Data Structures used by the Compilers and Assemblers. Symbol Table keeps into account the attributes of identifiers and other information. The attribute can be Type, Value, Scope and Address etc.

Q:-23) What is the use of System Call. (Dec. 2011)

Ans:-23) A **System call** is a mechanism used by an application for requesting a service from the operating system. Examples of the services provided by the operating system are allocation and deallocation of memory, reporting of current date and time etc. These services can be used by an application with the help of system calls.

Q:-24) What is Dynamic Linking. (Dec. 2011)

Ans:-24) Dynamic Linking is the mechanism by which Loading and Linking of external references are postponed until execution time. The main task of Loader is to Load the Main Program in Memory and the task of Linker is to Link the Sub Programs to the Main Program.

Q:-25) What is difference between Shell and a Kernel? (Dec. 2011)

Ans:-25) **Shell:** - A Shell is a software program that interprets commands from the user so that the OS can understand them and perform the appropriate functions. It's the outermost layer of a program.

Kernel is the essential center of a computer OS, the core that provides basic services for all other parts of the OS. It is the part of the OS that loads first, and it remains in main memory. Typically, the kernel is responsible for memory management, process and task management, and disk management.

Q:-26) What is Literal Table. (Dec. 2011)

Ans:-26) **Literal Table:-** It is used to store each Label and its corresponding value. The Literal Table created by Lexical Analysis to describe all Literals used in the Source Program.

Q:-27) What is the Role of Absolute Loader. (Dec. 2010)

Ans:-27) An **absolute loader** is the simplest type of loader scheme that fits the general model of loaders. The assembler produces the output in the same way as in the "compiler and go loader".

Disadvantage:

- The programmer has to specify the address to the assembler that where the program is to be loaded.
- It is very difficult to relocate in case of multiple subroutine.
- Programmer has to remember the address of each subroutine and use that absolute address explicitly in other subroutines to perform subroutine linkage

www.csepedia.com