SV.U.COLLEGE OF CM&CS

DEPARTMENT OF COMPUTER SCIENCE

MCA IV SEMESTER- II INTERNAL

Paper -System Programming

Time:40 minutes Marks: 20 Marks

1. The Linker is

1. Same as the loader
2. Required to create a load module
3. Always used before programs are executed
4. None

2. A system program that combines the separately compiled modules of a program into a form suitable for execution?

a. Assembler

b. Linking and Loader

c. Cross Compiler

d. Load and Go

3.A linker program

1. places the program in the memory for the purpose of execution.
2. relocates the program to execute from the specific memory area allocated to it.
3. links the program with other programs needed for its execution.
4. interfaces the program with the entities generating its input data.

4. Loading process can be divided into two separate programs, to solve some problems. The first is binder the other is ?

a.Linkage editor

b.Module Loader

c.Relocator

d.None of these

5.Resolution of externally defined symbols is performed by

a. Linker

b. Loader

c.Compiler

d.Editor

6. Relocatable programs

a. cannot be used with fixed partitions

b. can be loaded almost anywhere in memory

c. do not need a linker

d. can be loaded only at one specific location

7. Static memory allocation is typically performed during \_\_\_\_\_\_\_\_\_\_\_\_\_.

a. Compilation

b. Loading

c.Execution

d.Linking

8. If a number of instructions are repeating through the main program, then to reduce the length of the program\_\_\_\_\_\_is used

a)procedure  
b)subroutine  
c)macro  
d)none

9. The process of assigning a label or macroname to the string is called  
a)initialising macro  
b) initialising string macro  
c) defining a string macro  
d) defining a macro

10. A macro within a macro is called  
a) macro-within-macro  
b) nested macro  
c) macro-in-macro  
d) none of the mentioned

11. A macro can be defined as  
a) beginning of a program  
b) end of a program  
c) after initialisation of program  
d) anywhere in a program

12. A macro can be used as \_\_\_\_\_\_\_\_  
a) in data segment  
b) to represent directives  
c) to represent statements  
d) all of the mentioned

13. The end of a macro can be represented by the directive.  
a) END  
b) ENDS  
c) ENDM  
d) ENDD

14. Inserting the statements and instructions represented by macro, directly at the place of the macroname, in the program, is known as  
a) calling a macro  
b) inserting a macro  
c) initializing a macro  
d) none of the mentioned

15. The time required for execution of a macro is \_\_\_\_\_\_\_\_ that of the procedure.  
a) greater than  
b) less than  
c) equal to  
d) none of the mentioned

16. Which of the following statements is incorrect?  
a) complete code of instruction string is inserted at each place, wherever the macroname appears  
b) macro requires less time of execution than that of procedure  
c) macro uses stack memory  
d) macroname can be anything except registers and mnemonics

17. The beginning of the macro can be represented as  
a) START  
b) BEGIN  
c) MACRO  
d) None of the mentioned

18.A \_\_\_\_\_\_\_\_\_ is a software utility that translates code written in higher language into a low level language.

a. Converter

b. Compiler

c.Text Editor

d. Code optimizer

19. A bottom up parser generates  
a) Right most derivation  
b) Rightmost derivation in reverse  
c) Leftmost derivation  
d) Leftmost derivation in reverse

20. A grammar that produces more than one parse tree for some sentence is called  
a) Ambiguous  
b) Unambiguous  
c) Regular  
d) None of the mentioned

21. Parsing is also known as ?

a. Syntax analysis

b. Lexical analysis

c. Semantic analysis

d. None Of The Above

22. Bottom up parsing involves ?

a. shift reduce

b. Handle pruning

c. Operator check

d. A & B

23. Compiler translates the source code to

a. Executable code

b.Machine cde

c. Binary code

d. Both b and c

24. Which of the following groups is/are token together into semantic structures?

a. Syntax Analyzer

b. Intermediate generaton

c. Lexical analyzer

d. Semantic analyser

25. What is the output of lexical analyzer?

a. A parse tree

b. A list of tokens

c. Intermediate code

d. Machine code

26. Grammer of the programming is checked at \_\_\_\_\_\_\_ phase of compiler.

a. Semantic analysis

b. Syntax analysis

c.Code optimization

d. Code genearation

27. What is the action of parsing the source program into proper syntactic classes?

a. Lexical ana;ysis

b.Syntax analysis

c. General syntax analysis

d. Interpretation analysis

28. \_\_\_\_\_\_\_\_ is the most general phase structured grammar

a. Context sensitive

b.Regular

c.Context free

d. All of these

29. Lexical analysis is about breaking the sentence of characters into

a.Groups

b.Pockets

c.Lines

d.Tokens

30.Compiler can check\_\_\_\_\_\_ errors

a.Logical

b.Syntax

c.Content

d.All

31. The major number identifies the \_\_\_\_\_ associated with the device.  
a) driver  
b) protocol  
c) port  
d) none of the mentioned

32. The minor number range should be  
a) 0 to 15  
b) 0 to 63  
c) 0 to 255  
d) none of the mentioned

33. Which one of the following is not true?  
a) dynamic allocation of major numbers is not possible  
b) major number can not be shared among drivers  
c) dynamic allocation of major numbers is not possible & also major number can not be

shared among drivers  
d) none of the mentioned

34. In linux kernel 2.4, we can have  
a) 256 character drivers only  
b) 256 block drivers only  
c) 256 character drivers and 256 block drivers at the same time  
d) none of the mentioned

35. In we use a driver for N number of files, then we have to create \_\_\_\_ device files.  
a) N  
b) 1  
c) N-1  
d) None of the mentioned

36. If we use a driver for various device files, then  
a) minor number will be different for every device file  
b) minor number will be same for every device file  
c) minor number can not be allocated for any device file  
d) none of the mentioned

37. The connection between the device file and device driver is based on the  
a) name of device file  
b) number of device file  
c) name & number of device file  
d) none of the mentioned

38. In linux kernel 2.1, the minor numbers were used to  
a) represnt the sub-functionalitites of the driver  
b) identify the driver  
c) represent the device files  
d) none of the mentioned

39. The kernel identifies the driver with its  
a) module  
b) major number  
c) device file  
d) none of the mentioned

40. In linux, a device driver can work without the  
a) major number  
b) minor number  
c) device file name  
d) none of the mentioned

\*\*\*\*\*\*\*\*\*\*\*\*\*