

COMPUTER ENGINEERING DEPARTMENT

SUBJECT: SYSTEM PROGRAMMING & COMPILER CONSTRUCTION

COURSE: T.E.

YEAR: 2020-2021

SEMESTER: VI

DEPT: COMPUTER ENGINEERING

SUBJECT CODE: CSC602

EXAMINATION DATE: 04/06/2021

**SYSTEM PROGRAMMING & COMPILER CONSTRUCTION
ANSWER SHEET**

NAME : AMEY MAHENDRA THAKUR

SEAT NO. : 61021145

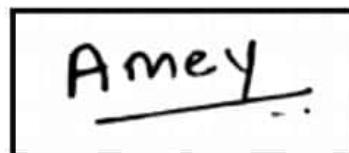
EXAM : SEMESTER VI

SUBJECT : SYSTEM PROGRAMMING & COMPILER CONSTRUCTION

DATE : 04-06-2021

DAY : FRIDAY

STUDENT SIGNATURE:



Q3.

A] i) Left Recursion.

Left Recursion

- A production of grammar is said to have left recursion if the leftmost variable of its RHS is same as Variable of its LHS.
- A grammar containing a production having left recursion is called as Left Recursive Grammar

Example:

$$S \rightarrow Sa / S$$

(Left Recursive Grammar)

- Left recursion is considered to be a problem situation for Top down parser
- Therefore, left recursion has to be eliminated from the grammar.

Elimination of Left Recursion

- Left recursion is eliminated by converting the grammar into a right recursive grammar.
- If we have the left-recursive pair of productions -

$$A \rightarrow A\alpha / \beta$$

(Left Recursive Grammar)

where β does not begin with A .

Then, we can eliminate left recursion by replacing the pair of productions with -

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' / \epsilon$$

(Right Recursive Grammar)

This right recursive grammar functions same as left recursive grammar.

Q3 A]

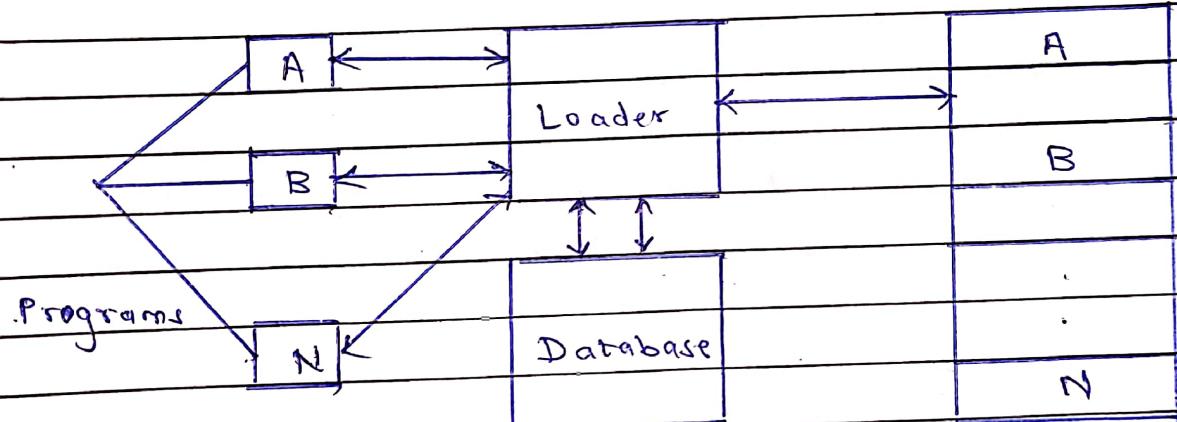
iii] System software vs Application software.

System software	Application software
① System software maintain the system resources and give the path for application software to run.	① Application software is built for specific task.
② Low level languages are used to write the system software	② While high level languages are used to write the application software.
③ Its a general purpose software	③ Its a specific purpose software
④ Without system software, system can't run.	④ Without application software system always runs
⑤ System software runs when system is turned on and stop when system is turned off	⑤ Application software runs as per the user's request.
⑥ Example: Operating system, etc	⑥ Example: Photoshop, VLC Player, etc
⑦ System software programming is complex than application software.	⑦ Application software programming is simpler as compared to system software.

Q.3.

B) i) Loader**Loader**

- A loader is a system program, which takes the object code of a program as input and prepares it for execution.
- Programmers usually define the program to be loaded at some predefined location in the memory.
- But this loading address given by the programmer is not be coordinated with the OS.
- The loader does the job of coordinating with the OS to get initial loading address for the .Exe file and load it into the memory.

Diagram**Functions of loader:**

- ① Allocation
- ② Linking
- ③ Relocation
- ④ Loading

Functions of loader.

① Allocation:

- Allocates the space in the memory where the object program would be loaded for execution.
- It allocates the space for program in the memory by calculating size of the program. This activity is called as allocation.
- In absolute loader allocation is done by the programmer and hence it is the duty of the programmer to ensure that the programs do not get overlap.
- In relocatable loader allocation is done by the loader. Hence the assembler must supply the loader the size of the program.

② Linking:

- It links two or more object codes and provides the information needed to allow references between them.
- It resolves the symbolic references (code / data) between the object modules by assigning all the user subroutine and library subroutine addresses. This activity is called linking.
- In absolute loader linking is done by the programmer as the programmer is aware about the runtime address of the symbol.
- In relocatable loader, linking is done by the loader and hence the assembler must supply to the loader, the locations at which the loading is to be done.

(3) Relocation:

- It modifies the object program by changing the certain instructions so that it can be loaded at different address from location originally specified.
- There are some address dependent locations in the program such address constants must be adjusted according to allocated space, such activity done by loader is called relocation.
- In absolute loader relocation is done by the assembler as the assembler is aware of the starting address of the program.
- In relocatable loader, relocation is done by the loader and hence assembler must supply to the loader the location at which relocation is to be done.

(4) Loading:

- It brings the object program into the memory for execution.
- Finally it places all the machine instructions and data of corresponding programs and subroutines into the memory. Thus program now becomes ready for execution, this activity is called loading.
- In both the loaders (absolute, relocatable) loading is done by the loader and hence the assembler must supply to the loader the object program.

Dynamic Loading Loader:

- If all subroutines are loaded simultaneously into the core, then there may be chance, that the core available may be insufficient for the subroutines, this may lead to further complications.
- Execution of such program can be possible if all the segments are not required simultaneously to be present in the main memory.
- In such situations only those segments are residents in the memories that are actually needed at the time of execution.
- But the question arises what will happen if the required segment is not present in the memory.
- Naturally the execution process will be delayed until the required segment gets loaded in the memory.
- Overlay structure is used to specify the inter dependency between all the segments.
- It consists of nodes and edges and segment is represented by the nodes.
- If the two are on the same path they can lie in the main memory.
- For solving such problems techniques like segmentation and paging is used.
- In these the subroutines are loaded into core at a different time because the subroutines in a program are needed at different times. i.e. they may be mutually exclusive, by identifying which subroutine can call other subroutines to produce an overlay structure.

Overlay structure is used to:

- ① Keep in memory only those instructions and data that are needed at any given time.
- ② Overlay structure needed when process is larger than amount of memory allocated to it.
- ③ Implemented by user, no special support needed from operating system. Programming design of overlay structure is complex.

The flipper or overlay supervisor is the portion of the loader that actually intercepts the "calls" and loads the necessary procedure.

Advantages:

- ① No wastage of memory.
- ② A system can be dynamically loaded whenever it is required.

(Q.3 B) ii]

Loop Optimization Techniques

① Code movement:

- As the name suggests, it involves movement of the code.
- The code present inside the loop is moved out if it does not matter whether it is present inside or outside.
- Such a code unnecessarily gets execute again and again with each iteration of the loop.
- This leads to the wastage of time at run time.

② Strength reduction:

- As the name suggests, it involves reducing the strength of expression.
- This technique replaces the expensive and costly operators with the simple and cheaper ones.

③ Frequency reduction:

- As the name suggests, it involves reducing the frequency of expression.

④ Loop distribution:

- Loop distribution attempts to remove the sequential statement into a separate loop and gather the parallelizable statements into a different loop.