
**MUMBAI UNIVERSITY STUDENTS ASSOCIATION (MUSA)**

**VIVA QUESTION AND ANSWER OF SPCC**

**S.E SEM-VI** **BRANCH: COMPS** **FOR SUMMER SESSION 2025**

---

## VIVA QUESTIONS AND ANSWERS

### Module 1: Introduction to System Software

1. **Q:** What is system software?
   **A:** System software is a type of software that manages hardware and provides services for application software.
2. **Q:** Give two goals of system software.
   **A:** Efficiency and abstraction from hardware.
3. **Q:** What is system programming?
   **A:** System programming involves writing software that interacts closely with hardware or other system software.
4. **Q:** Name any three types of system programs.
   **A:** Assembler, Compiler, Linker.
5. **Q:** What is the difference between a compiler and an interpreter?
   **A:** A compiler translates the entire program before execution, while an interpreter translates line-by-line during execution.
6. **Q:** What is the role of an operating system as system software?
   **A:** It manages resources and provides a user interface to interact with hardware.
7. **Q:** What does a macro processor do?
   **A:** It replaces macros with their definitions during preprocessing.
8. **Q:** Define a loader.
   **A:** A loader loads executable code into memory for execution.
9. **Q:** What is a debugger used for?
   **A:** It helps in finding and fixing errors in a program.
10. **Q:** What does an editor in system software do?
    **A:** It allows writing and modifying source code.

### Module 2: Assemblers

1. **Q:** What is assembly language?
   **A:** A low-level programming language that uses mnemonics instead of binary code.
2. **Q:** Define an assembler.
   **A:** An assembler converts assembly language code into machine code.

3. **Q:** What is the pass structure of an assembler?
   **A:** It refers to the number of times the source code is read to generate machine code (e.g., single-pass or two-pass).
4. **Q:** What happens in the first pass of a two-pass assembler?
   **A:** It creates a symbol table and handles labels.
5. **Q:** What is the main task of the second pass of a two-pass assembler?
   **A:** Generates final machine code using the symbol table.
6. **Q:** Define a symbol table.
   **A:** A table storing label names and their addresses.
7. **Q:** What are the main data structures used in assembler design?
   **A:** Symbol table, literal table, and opcode table.
8. **Q:** What is a literal in assembly language?
   **A:** A constant value used in the program.
9. **Q:** What is the difference between single-pass and two-pass assembler?
   **A:** A single-pass assembler processes the source in one go, while two-pass does it in two phases for better forward reference handling.
10. **Q:** What is an opcode table used for?
    **A:** To map mnemonics to their corresponding machine instructions.

## Module 3: Macros and Macro Processor

1. **Q:** What is a macro?
   **A:** A single instruction that expands into a set of instructions.
2. **Q:** What does a macro processor do?
   **A:** It processes macros before actual assembly.
3. **Q:** What is a parameterized macro?
   **A:** A macro that accepts arguments during its call.
4. **Q:** What is the purpose of using macros?
   **A:** To reduce repetitive code and simplify programming.
5. **Q:** What is a conditional macro?
   **A:** A macro that can generate different outputs based on a condition.
6. **Q:** What is macro expansion?
   **A:** Replacing the macro call with its corresponding definition.
7. **Q:** What is a nested macro?
   **A:** A macro that contains another macro inside its definition.
8. **Q:** What are the two passes in macro processing?
   **A:** Pass 1: Macro definition table is created. Pass 2: Macro calls are expanded.
9. **Q:** Name two data structures used in macro processing.
   **A:** Macro Definition Table (MDT), Macro Name Table (MNT).
10. **Q:** How is a macro different from a subroutine?
    **A:** A macro is expanded inline; a subroutine is called during execution.

## Module 4: Loaders and Linkers

1. **Q:** What is a loader?
   **A:** A loader loads the compiled program into memory for execution.
2. **Q:** Define a linker.
   **A:** A linker connects different object modules and resolves references.
3. **Q:** What is relocation in loading?
   **A:** Modifying object code addresses to match memory locations.
4. **Q:** What is linking?
   **A:** Resolving external references between object files.
5. **Q:** What is a relocating loader?
   **A:** A loader that adjusts addresses during program loading.
6. **Q:** What is a direct linking loader?
   **A:** A loader that links and loads the program at the same time.
7. **Q:** What is dynamic linking?
   **A:** Linking external modules during program execution.
8. **Q:** What is dynamic loading?
   **A:** Loading modules into memory only when needed at runtime.
9. **Q:** Give an advantage of dynamic linking.
   **A:** Reduces memory usage by sharing libraries among programs.
10. **Q:** What is absolute loading?
    **A:** Loading a program at a fixed memory address without relocation.

## Module 5: Compiler - Analysis Phase

1. **Q:** What is a compiler?
   **A:** A software that converts high-level language into machine code.
2. **Q:** Name the phases of a compiler.
   **A:** Lexical, Syntax, Semantic, Intermediate Code Gen, Optimization, Code Gen.
3. **Q:** What is lexical analysis?
   **A:** The phase that converts source code into tokens.
4. **Q:** Which tool is used in lexical analysis?
   **A:** Finite State Automata (FSA).
5. **Q:** What is syntax analysis?
   **A:** Checks if the tokens follow correct grammar using parse trees.
6. **Q:** What is the role of context-free grammar?
   **A:** Defines the syntax rules for the programming language.
7. **Q:** Name two types of parsers.
   **A:** Top-down (LL(1)), Bottom-up (Shift Reduce, Operator Precedence).

8. **Q:** What is semantic analysis?
   **A:** Checks the meaning and correctness of code using syntax-directed definitions.
9. **Q:** What is a parse tree?
   **A:** A tree structure showing how the source code conforms to grammar.
10. **Q:** What is SLR parsing?
    **A:** A simple bottom-up parsing technique using LR(0) items and follow sets.

**Module 6: Compiler - Synthesis Phase**

1. **Q:** What is intermediate code generation?
   **A:** The phase that converts syntax tree to an intermediate representation.
2. **Q:** Name types of intermediate code.
   **A:** Syntax Tree, Postfix Notation, Three Address Code (TAC).
3. **Q:** What is Three Address Code?
   **A:** Intermediate code where each instruction has at most three operands.
4. **Q:** What is a triple in TAC?
   **A:** An ordered set of operation and operands stored using position/index.
5. **Q:** What is a quadruple?
   **A:** A 4-field structure: Operator, Argument1, Argument2, Result.
6. **Q:** What is an indirect triple?
   **A:** A table of pointers pointing to triples to allow code reordering.
7. **Q:** Why is code optimization needed?
   **A:** To improve performance and reduce code size.
8. **Q:** Name two types of code optimization.
   **A:** Machine Dependent and Machine Independent.
9. **Q:** What is a basic block?
   **A:** A sequence of statements with one entry and one exit point.
10. **Q:** What is a flow graph?
    **A:** A graphical representation showing flow of control between basic blocks.

*******ALL THE BEST*******