|  |
| --- |
| **Scheme of Valuation** |
| **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**FIFTH SEMESTER(S) B.TECH DEGREE EXAMINATION, MAY 2019 |
| **Course Code: CS303** |
| **Course Name: SYSTEM SOFTWARE** |
| Max. Marks: 100 |  | Duration: 3 Hours |
| **PART A** |
|  |  | ***Answer all questions, each carries3 marks.*** | Marks |
| 1 |  | Explain three functions of Operating System Ans: Resource management, Memory Management, Process Management, Storage management, I/O Managenemt, Provide environment for running programs, etc. Write any three (3 x 1 = 3 marks) | (3) |
| 2 |  | Write a sequence of instructions for SIC/ XE to find the average of three numbers ,BETA, GAMMA and DELTA.Ans: Use ADD/ ADDR and DIV or DIVR Instructions. (3 marks) | (3) |
| 3 |  | Explain the format of the object program generated by a two-pass SICAssembler, highlighting the contents of each record type Ans: Header, Text and End Records. (1 x 3 = 3 marks) | (3) |
| 4 |  | Explain the data structures used and their purposes in a two-pass assembler Ans: Symbol Table: SYMTAB- Contain name and value (address) of each symbol (1.5 marks)Operation Code Table: OPTAB – Contains Opcodes (1.5 marks) | (3) |
|  |  |  |  |
| **PART B** |
| ***Answer any two full questions, each carries9 marks.*** |
| 5 |  | Compare the features of Standard SIC and SIC/XE architecture.Ans: MemoryData Formats Explain each.Instruction Formats 9 MarksInstruction SetInput and Output | (9) |
| 6 | a) | Explain assembler directives. List any four assembler directives in SICmachine.Ans: Assembler Directives are instruction to assembler (1 Mark)START, END, BYTE, WORD, RESB, RESW etc. Write any four. (1 x 4 =4 marks) | (5) |
|  | b) | Explain the concept of program relocation with an example.Definition:Moving/ Loading program to a different location other than specified. 2 MarksExample/ Figure : 2 marks | (4) |
| 7 |  | Write the algorithms for Pass 1 and Pass 2 of a two-pass assembler Ans: Pass 1: Entry into SYMTAB & OPTAB and resolving forward references- Give the detailed steps (4.5 marks)Pass 2: Actual translation to machine code –Give the detailed steps (4.5 marks) | (9) |
|  |  |  |  |
| **PART C** |
| ***Answer all questions, each carries3 marks.*** |
| 8 |  | Differentiate Define record and Refer record.Ans: Define Record: To record External Definitions ( EXTDEF)- Explanation (1.5 marks)Refer Record: To record External References (EXTREF)- Explanation (1.5 marks) | (3) |
| 9 |  | Explain how forward references are resolved during program assembling in a single pass assembler.Ans: 1) Load-and-go one-pass assembler - Explain(1.5 marks)2) Generate another Text record with correct operand address – Explain ( 1.5 marks) | (3) |
| 10 |  | Give the absolute loader algorithm Ans: Absolute loader: Only loading, no linking and relocation.Algorithm of an absolute loader (3 marks) | (3) |
| 11 |  | Explain the concept of Automatic Library Search.Ans: Subroutines are automatically fetched from the library, linked and loaded.(3 mark) | (3) |
|  |  |  |  |
| **PART D** |
| ***Answer any two full questions, each carries9 marks.*** |
| 12 |  | Differentiate Program Blocks and Control Sections. Explain how address calculation is performed in the case of Program Blocks. Ans:Program Block and Control Section Definition (2 marks)Address Calculation in Program Blocks (7 marks) | (9) |
| 13 | a) | Explain the working of Multipass Assemblers with an exampleExplanation: (3 marks)Example : (2 marks) | (5) |
| 13 | b) | Explain Dynamic Linking with an example.Ans :Explanation :Postpones linking function until execution ( marks)Example : (2 Marks) | (4) |
| 14 |  | Which are the data structures used during the operation of a linking loader? Write the algorithm for Pass 2 of a Linking Loader Ans: Data structures used (4 marks) Algorithm of Pass 2 of Linking Loader (5 Marks) | (9) |
|  |  |  |  |
| **PART E** |
| ***Answer any four full questions, each carries10 marks.*** |
| 15 | a | A code segment need to be repeatedly used in various parts of assembly language program and fast execution is also needed. Would you use a macro or a subroutine? Justify your answer with help of examples. Ans: Macro is suited best (2 marks)Explanation with examples (3 marks) | (5) |
|  | b | List and explain the different design options available for macroprocessors.Three design options – 5 marks | (5) |
| 16 |  | Certain macroprocessor features are independent of the machine architecture. Give the details of such machine independent macro-processor features.Ans: Concatenation of Macro parameters (2.5 Marks)Generation of Unique Labels (2.5 Marks)Conditional Macro Expansion (2.5 Marks)Keyword Macro parameters (2.5 Marks) | (10) |
| 17 |  | Write the algorithm for one pass macro processor and explain the process, showing when and how the different data structures are used.Ans: Algorithm (6 marks)Explanation - 4 marks | (10) |
| 18 |  | Using a neat diagram, explain the structure of a text editor.Ans: Figure: (3 marks)Explanation (7 marks) | (10) |
| 19 |  | A new hardware device is plugged into a system. Which is the appropriate system software needed for the proper working of the new hardware? Give its functionalities and general architecture. Ans: The appropriate system software is Device Driver (2 marks)Functionalities& architecture (8 marks) | (10) |
| 20 |  | Write down the situations where debugging by induction, deduction and backtracking are used, explaining each process. Ans:  Explanation (10 marks)Induction debugging start with the symptoms of the error, possibly in the result of one or more test cases, and looking for relationships among the symptoms, the error is often uncovered.The process of deduction proceeds from some general theories or premises, using the processes of elimination and refinement, to arrive at a conclusion (the location of the error).Debugging by Backtracking: An effective method for locating errors in small programs is to backtrack the incorrect results through the logic of the program until you find the point where the logic went astray. In other words, start at the point where the program gives the incorrect result—such as where incorrect data were printed. At this point you deduce from the observed output what the values of the program's variables must have been. By performing a mental reverse execution of the program from this point and repeatedly using the process of "if this was the state of the program at this point, then this must have been the state of the program up here," you can quickly pinpoint the error.  | (10) |
| \*\*\*\* |