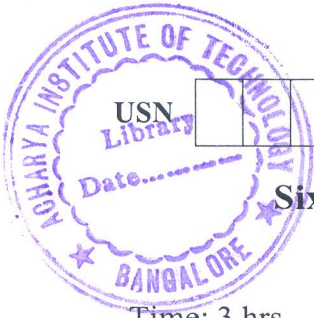


CBCS SCHEME



15CS63

Sixth Semester B.E. Degree Examination, Aug./Sept.2020 System Software and Compiler Design

Time: 3 hrs.

Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

- 1 a. Explain SIC Architecture in detail. (06 Marks)
- b. Explain the data structures and Pass-1 algorithm of SIC/ME assembler. (06 Marks)
- c. List out the differences between system software and application software. (04 Marks)

OR

- 2 a. List the different addressing modes used in SIC/XE. Give instructions format for each and explain the addressing mode. (08 Marks)
- b. Generate the machine code for the following :
 - (i) 0000 +JSUB RDREC
 - (ii) 0004 STL RETADR
 - (iii) 0008 LDB #LENGTH
 - (iv) 000A CLEAR X

Assume the opcodes are:

JSUB = 48_H, STL = 14_H, LDB = 60_H, CLEAR = B4_H

The LC value for : RDREC = 1036_H

RETADR = 0030_H, LENGTH = 0033_H

The mnemonics values for registers are

A = 0, X = 1, L = 2, B = 3, S = 4, T = 5, F = 6, Pc = 8, SW = 9.

(08 Marks)

Module-2

- 3 a. With an example show how relocation and linking operations are performed. (08 Marks)
- b. With source code, explain the working of boot-strap loader. (08 Marks)

OR

- 4 a. Explain machine independent loader features given an example with implementation. (08 Marks)
- b. With a neat diagram, explain how object program can be processed using linkage loader and linkage editor. (08 Marks)

Module-3

- 5 a. With a neat diagram explain the different phases of the compiler. (10 Marks)
- b. Explain the concept of input buffering with its implementation. (06 Marks)

OR

- 6 a. Describe language processing system with a neat diagram. (06 Marks)
- b. Write the transition diagram for the following :
 - (i) relop
 - (ii) unsigned numbers
 - (iii) identifiers(06 Marks)
- c. Differentiate between compiler and interpreter. (04 Marks)

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg. 42+8 = 50, will be treated as malpractice.

Module-4

- 7 a. Compute : (i) First() and Follow()
 (ii) Predictive parsing table for the given grammar
 $D \rightarrow L ; T$
 $L \rightarrow L ; id | id$
 $T \rightarrow int | real$ (06 Marks)
- b. Consider the CFG with the production set,
 $E \rightarrow E + T | T$
 $T \rightarrow TF | F$
 $F \rightarrow F* | a | b$
- Compute the following,
 (i) FIRST() and FOLLOW()
 (ii) Set of LR(0) items
 (iii) SLR parsing table. (10 Marks)

OR

- 8 a. Compute the following for the given grammar.
 $S \rightarrow AA$
 $A \rightarrow a | b$
- (i) LR(1) items (ii) Canonical Parsing table (iii) Verify for any valid string. (10 Marks)
- b. Write a short note on shift reduce parsing with an example. (06 Marks)

Module-5

- 9 a. Write the annotated parse tree and its syntax directed definition to obtain
 $1 * 2 * 3 * (4 + 5)^n$ for the grammar
 $L \rightarrow En$
 $E \rightarrow E + T | T$
 $T \rightarrow T * F | F$
 $F \rightarrow (E) | digit$ (06 Marks)
- b. Translate the arithmetic expression :
 $a * -(b + c)$ into
 (i) Quadruples (ii) Triples (iii) Indirect triples (06 Marks)
- c. Discuss various issues in the design of code generation. (04 Marks)

OR

- 10 a. By considering an array type $int[3][3]$, write syntax directed translation with semantic rules and its annotated parse tree. (06 Marks)
- b. Obtain the directed acyclic graph for the expression $x + x * (y + z) + (y + z) * w$, along with the steps. (06 Marks)
- c. Generate assembly level language code (target code) for the following three address sequence assuming that p and q are in memory locations:
 $y = *q$
 $q = q + 4$
 $*p = y$
 $p = p + 4$ (04 Marks)
