

CBCS Scheme

USN

--	--	--	--	--	--	--	--	--	--

16MCA25

Second Semester MCA Degree Examination, June/July 2017 System Software

Time: 3 hrs.

Max. Marks: 80

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

Module-1

- 1 a. Compare system software and application software. Give examples for each. (04 Marks)
- b. With reference to SIC/XE machine architecture, explain instruction formats and addressing modes clearly indicating the settings of different flag bits. (08 Marks)
- c. Give the target address generated for the following instructions (hexadecimal), if (B) = 006000, (PC) = 003000, (X) = 000090 : i) 75101000, ii) 032026. (04 Marks)

OR

- 2 a. Explain the following with an example for each: i) WORD, ii) START. (04 Marks)
- b. With reference to VAX machine architecture, explain memory, registers, data formats and instruction formats. (08 Marks)
- c. ALPHA is an array of 100 words. Write a sequence of instructions for SIC/XE to set all 100 elements of the array to 0. (04 Marks)

Module-2

- 3 a. Distinguish between a literal and an immediate operand with an example for each. (04 Marks)
- b. Generate the complete object program for the following assembly language program, clearly showing the symbol table entries. All addresses are in hexadecimal.
Assume: LDT - 74, LDX - 04, LDCH - 50, STCH - 54, TIXR - B8, JLT - 38, X - 1, T - 5.
ASCII character codes (decimal): E - 69, O - 79, F - 70.

```
COPY      START  0
FIRST     LDT    #3
          LDX    #0
MOVECH    LDCH   STR1, X
          STCH   STR2, X
          TIXR   T
          JLT   MOVECH
STR1      BYTE   C 'EOF'
STR2      RESB   3
          END    FIRST
```

- c. Explain the features of MASM assembler. (04 Marks)

OR

- 4 a. What is a relocatable program? Explain the concept of program relocation with an example, and the means for implementing it. (04 Marks)
- b. What are program blocks? Mention the relevant assembler directives used in writing SIC/XE source program involving program blocks and hence briefly discuss how are they handled by an assembler. Give example. (08 Marks)
- c. Compare a two-pass assembler with a one-pass assembler. Bring out the difference involved in handling forward references. (04 Marks)

Module-3

- 5 a. Briefly explain a simple boot strap loader with an algorithm or a source program. (08 Marks)
 b. Distinguish between a linkage editor and a linking loader. (05 Marks)
 c. Enlist any three loader options specified using a command language. (03 Marks)

OR

- 6 a. Discuss the detailed design of a linking loader with an example. (08 Marks)
 b. Explain dynamic linking with suitable diagrams. (05 Marks)
 c. Enlist the different types of SunOS linkers and the associated output modules produced. (03 Marks)

Module-4

- 7 a. Mention the basic functions of a macroprocessor. (03 Marks)
 b. Discuss with a suitable example, the usage of various data structures in handling an assembly language program involving macros. (08 Marks)
 c. Explain the ANSI C macro language with examples. (05 Marks)

OR

- 8 a. With an example briefly explain keyword macro parameters. (03 Marks)
 b. Expand the following macro invocation statements using the macro definition given below.
 i) RDBUFF F1, BUFFER, (04, 12), LENGTH
 ii) RDLOOP RDBUFF F2, BUFF, , LEN

```

RDBUFF MACRO &INDEV, &BUFADR, &EOR, &RECLTH
&EORCT SET %NITEMS (&EOR)
        CLEAR X
        CLEAR A
        +LDT #4096
$LOOP TD =X '&INDEV'
        JEQ $LOOP
        RD =X '&INDEV'
&CTR SET 1
        WHILE (&CTR LE &EORCT)
        COMP = X '0000&EOR[&CTR]'
        JEQ $EXIT
&CTR SET &CTR + 1
        ENDW
        STCH &BUFADR, X
        TIXR T
        JLT $LOOP
$EXIT STX &RECLTH
        MEND

```

- (08 Marks)
 c. Mention the advantages of general purpose macro processors. Discuss the details that must be considered while designing a general purpose macroprocessor. (05 Marks)

Module-5

- 9 a. What is a grammar? Using the BNF grammar below, represent the syntax analysis of the PASCAL statement VAR := SUMSQ DIV 100 – MEAN * MEAN in the form of a parse tree.

$\langle \text{assign} \rangle ::= \text{id} := \langle \text{exp} \rangle$
 $\langle \text{exp} \rangle ::= \langle \text{term} \rangle \mid \langle \text{exp} \rangle + \langle \text{term} \rangle \mid \langle \text{exp} \rangle - \langle \text{term} \rangle$
 $\langle \text{term} \rangle ::= \langle \text{factor} \rangle \mid \langle \text{term} \rangle * \langle \text{factor} \rangle \mid \langle \text{term} \rangle \text{DIV} \langle \text{factor} \rangle$
 $\langle \text{factor} \rangle ::= \text{id} \mid \text{int} \mid (\langle \text{exp} \rangle)$

(07 Marks)

- b. Explain the various types of machine-independent code optimization techniques. (05 Marks)
 c. Indicate whether the finite automation given in Fig.Q9(c) recognizes the following strings.
 i) 9Alpha ii) Num_2 iii) Hello__world iv) bbb_9_

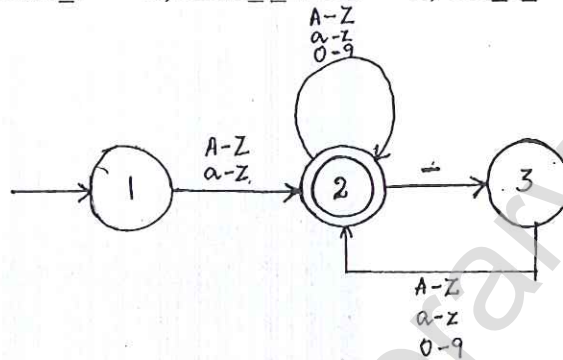


Fig.Q9(c)

(04 Marks)

OR

- 10 a. Write the recursive-descent parse for a READ statement and show the corresponding syntax tree constructed for the statement READ (VALUE). The BNF grammar is given by the following:

$\langle \text{read} \rangle ::= \text{READ} (\langle \text{id_list} \rangle)$
 $\langle \text{id_list} \rangle ::= \text{id} \{ , \text{id} \}$

(07 Marks)

- b. Explain P-code compilers with a neat diagram. (05 Marks)
 c. Assume the array A is declared A:ARRAY{1..5, 1..6} OF INTEGER with each element occupying 3 bytes. Generate quadruples for the statement A[I, J] := 0. (04 Marks)

CMRIT Library