

ALCCS

Code: CS22
Time: 3 Hours

Subject: SYSTEM SOFTWARE
Max. Marks: 100

SEPTEMBER 2010

NOTE:

- Question 1 is compulsory and carries 28 marks. Answer any FOUR questions from the rest. Marks are indicated against each question.
 - Parts of a question should be answered at the same place.
-

- Q.1** a. Explain the various data structures required by pass-1 of assembler?
- b. Differentiate between macro call and procedure call.
- c. List four software tools that assist a programmer during program testing and debugging.
- d. What is dynamic linking? List its advantages.
- e. What are machine-independent features in assembler design?
- f. Design a DFA for recognizing identifiers, unsigned integers and unsigned real numbers with fractions.
- g. What is a macro processor? (7 × 4)

- Q.2** a. For the program shown below write the quadruples.

```
PROGRAM STATS
VAR
    SUM, SUMSQ, I, VALUE, MEAN, VARIANCE: INTEGER
BEGIN
    SUM := 0 ;
    SUMSQ := 0 ;

    FOR I := 1 to 100 DO
        BEGIN
            READ (VALUE) ;
            SUM := SUM + VALUE ;
            SUMSQ := SUMSQ + VALUE * VALUE
        END;
    MEAN := SUM DIV 100;
    VARIANCE := SUMSQ DIV 100 - MEAN * MEAN ;
    WRITE (MEAN, VARIANCE)
END.
```

- b. Describe various parameter passing mechanisms. (12+6)

- Q.3** a. Explain the various techniques to identify free memory areas as a result of allocation and de-allocations in a heap.
- b. Define the following:

- (i) Formal language Grammars.
- (ii) Terminal symbols.
- (iii) Alphabet and String.

(9+9)

Q.4 a. What is an LL(1) parser? Is there any advantage of using LL(1) parsing?
parser for the following grammar

Construct a parser table for an LL(1)

$E ::= TE$
 $E ::= +TE \mid \varepsilon$
 $V ::= VT$
 $T ::= *VT \mid \varepsilon$
 $V ::= <id>$

b. Grammar containing left recursion is not amenable to top down parsing. Comment and explain giving a suitable example. **(12+6)**

Q.5 a. What is an assembly language? What are the advantages of using assembly language? Why do you need assembler directives-EQU and ORIGIN? Discuss with suitable example.

b. Write short notes on: (i) Debug monitors (ii) User Interfaces

(10+8)

Q.6 a. Explain Lexical substitution during macro expansion.

b. Write an algorithm to outline the macro-expansion using macro-expansion counter.

c. Write a recursive macro to reverse the characters of a string.

(6+6+6)

Q.7 a. Differentiate between

- (i) Program translation and program interpretation.
- (ii) Absolute loader and relocating loader
- (iii) Problem-oriented and procedure-oriented language

b. What do you mean by code optimization? What is the aim of this phase? Explain elimination of common subexpressions during code optimisation. **(12+6)**