

ANNA UNIVERSITY COIMBATORE

B.E. / B.TECH. DEGREE EXAMINATIONS : DECEMBER 2009

REGULATIONS - 2007

FOURTH SEMESTER – COMPUTER SCIENCE & ENGINEERING

070230044 - SYSTEM SOFTWARE

TIME : 3 Hours

Max.Marks : 100

PART – A

(20 x 2 = 40 MARKS)

ANSWER ALL QUESTIONS

1. Differentiate a CISC and a RISC computer.
2. Name the two phases of compilation process.
3. In which way, assembly level language simplifies the job of a machine level language?
4. Specify the usage of interpreter in programming execution.
5. Give one example on literal.
6. List the major structures present in an assembler.
7. Write down the input and outcome of one-pass assembler.
8. Draw a model format of symbol table used in assemblers.
9. Distinguish absolute and relocatable loaders with respect to efficiency.
10. Trace one demand that motivates dynamic linking.
11. List the limitations of bootstrap loader.
12. How do you encourage balancing of load through loaders? Discuss.
13. Give the common algorithmic syntax of a macro.
14. Locate one example for Macro within Macro.
15. Define the limitations of ANSI C Macro language.
16. Can we concatenate two Macro parameters? Explain.
17. List any two challenges of defining a user interface.

18. Draw the component level architecture of a text editor.
19. Define the process of debugging
20. List any two tools used for debugging.

PART - B

(5 x 12 = 60 Marks)

ANSWER ANY FIVE QUESTIONS

21. Categorize and provide suitable examples on addressing modes.
22. Write a pseduocode on two pass assembler and discuss the intermediate stages.
23. Locate the efficiency achieved by MASM assembler with respect to a tracing example.
24. Draw a data flow diagram to depict the steps involved in dynamic linking.
25. Explain how relocatable loader is employed over the process with one suitable example.
26. Brief the concept of MASM macro processor and provide suggestions to improve its performance.
27. Design an user interface for sales-inventory control domain and analyze its pros and cons.
28. Discuss any four practical challenges of debugging process with suitable examples.