

1. The larger memory available on SIC/XE means that an address will (in general) no longer fit into a 15-bit field; thus the instruction format used on the standard version of SIC is no longer suitable. Please suggest two solutions to solve the problem and briefly describe them. (8%)
2. What is program relocation? What kind of information should be provided by the assembler for the loader in order to solve program relocation? (8%)
3. 爲何一般 assembler 的設計是採用相對定址(Relative Addressing)而不採用絕對定址? 絕對組譯(Absolute Assembly)和相對組譯(Relative Assembly)程式各在何時(Assembly time, Load time, Link time, Execution time)決定載入記憶體的位置。(8%)
4. Consider the following preemptive priority-scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue, but not running), its priority changes at a rate α ; when it is running, its priority changes at a rate β . All processes are given a priority of 0 when they enter the ready queue. The parameters α and β can be set to give many different scheduling algorithms.(8%)
 - (i) FCFS algorithm results from (a) $\alpha > \beta > 0$ (b) $\alpha < \beta < 0$ (c) $\beta > \alpha > 0$ (d) $\beta < \alpha < 0$
 - (ii) LIFO algorithm results from (a) $\alpha > \beta > 0$ (b) $\alpha < \beta < 0$ (c) $\beta > \alpha > 0$ (d) $\beta < \alpha < 0$
5. Suppose we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds? (8%)
6. Please describe the UNIX file system data structures. Also please don't describe it using so-called tree structures. (10%)
7. Describe the necessary data structures and the algorithm for a 1-pass macro processor that allow macro definitions within macros? (10%)
8. What is dynamic linking loader? What advantage does dynamic linking provide? (10%)
9. Construct an LL(1) table for the following grammar: (10%)
 - (a) $S \rightarrow A\$$
 - $A \rightarrow Ap \mid Bq$
 - $B \rightarrow rB \mid \lambda$

If the grammar is not LL(1), transform it into an equivalent LL(1) grammar first.
10. We can prove that two regular expressions are equivalent by showing that their minimum-state DFA's are the same, except for state names. Using this technique, show that the following regular expressions are all equivalent. (10%)
 - (a) $(a \mid b)^*$
 - (b) $(a^* \mid b^*)^*$
11. Is it possible to write a context-free grammar which generates all nonempty strings with equal numbers of a's, b's, c's in alphabetical order? Why? (10%)