國立成功大學 87 學年度 工程科學研究所 考試(系統程式 試題) 共 2 頁 第 1 頁

1. Consider a general two-pass assembler and two-pass linking loader. Identify the passes of the assembler or loader where each of the following would be done:

(a) Assign relative addresses to symbols defined as labels in the program.
(b) Load the object program into memory.
(c) Assign actual addresses to program modules.
(d) Generate the object program.
(e) Modify the relocatable portion of the object code.
(f) Enter address for external symbols into the external symbol symbol.

(10%)

2. In general, a solution to the critical-section problem must satisfy three requirements: mutual exclusion, progress, and bounded waiting.

(a) Briefly describe these requirements.
(b) Evaluate the following two-process mutual exclusion algorithm in terms of the three requirements for such algorithms. Which requirements are satisfied? Which are not satisfied?
(c) Under what situations, "busy waiting" for entering Critical section has its advantage.

```
While (1)
{
  flag[i] = TRUE;
  while (flag[j] == TRUE)
  {
    if (turn == j)
    {
      flag[i] = FALSE;
      while (turn == j);
      flag[i] = TRUE;
    }
  }

    Critical Section

  turn = j;
  flag[i] =FALSE;
}
```

(10%)

3. Consider First-fit, Best-fit, and Worst-fit allocation algorithm:
(a) Given memory partitions of 100K, 500K, 200K, 300K, and 600K (in order), how would each of these algorithms place processes of 212K, 417K, 112K, and 426K (in order)?
(b) Let Tb be the average time required to allocate a block of memory using the Best-Fit. Similarly, Tw and Tf are the average times required when using Worst-fit and First-fit, respectively. Show the relations among Tb, Tw, and Tf in terms of mathematical expression(s). Explain your answers.

(10%)

4. The following questions deal with the Banker's Algorithm deadlock avoidance. Be sure to provide a reasonable explanation for your answers.
(a) Consider an enhancement to the resource management system that would allow a process to increase its maximum resource need vector at an arbitrary time after beginning execution. Under what conditions, if any, should the resource manager allow a process to do so?
(b) Suppose there exists a resource type that represents tape drives available for use by processes. If one of the currently unused tape drives breaks down, what effects, if any, will this have on the resource allocation state?

(10%)

5. The following is an access (or protection) matrix:

domain \ object	File-1	File-2	File-3	File-4	Printer-1	Card Reader
1	write	read			read	
2		write	read execute	read write	write	
3						write

(a) Point out any possible errors for above table. if there is no error then write OK for your answer.
(b) Show the capability list (table style) for domain 2.
(c) Show possible protection methods for capability list.
(d) If the table is changed into dynamic protection structure, show all additional access rights.
(10%)

6. Consider one-pass assembler and one-pass linking loader:
(a) The process of fixing up a few forward references should involve less overhead than making a complete second pass of the source program. Why don't all assemblers use the one-pass technique for efficiency?
(b) Suggest a design for a one-pass linking loader What restrictions (if any) would be required? Describe the advantages and disadvantages of such a one-pass loader.
(10%)

7.
(a) What are the differences between Macro and Subprogram (i.e. Function or Subroutine).
(b) How to solve the problems of label and parameter, when doing Macro expansion.
(10%)

8. Consider the following two expressions:

(1) x(i, j), (2) y(a, b)

x(i, j) is array reference, while y(a, b) is function call.
(a) Define a grammar for each of them.
(b) Based on your grammar of (a), describe how to recognize them when you write a compiler.
(10%)

9. Consider grammar: E --> E + E | E * E | (E) | -E | id
(a) Does above grammar belong to operator precedence grammar? If yes, write its operator precedence matrix.
(b) Does it belong to ambiguous grammar? If yes, how to transform it into an unambiguous gramar?
(10%)

10. Design a method to manage register. The method can prevent the generation of consecutive commands of:

```
Store A
Load  A
```

while doing code generation. Explain how you achieve it.
(10%)