⑯ 學年度 國立成功大學 碩士班招生考試 資訊工程所 系統程式 (乙組) 試題 共二頁 第一頁

Part I. Operating Systems (50%)

1. (15%) State the design schemes for CPU scheduler, virtual memory and other related aspects of a general-purpose operating system that supports (soft) real-time computing.

2. (15%) Describe in proper detail all of the feasible mechanisms (including low-level, high-level, hardware and software strategies) to achieve process synchronization. To each scheme state its defects/merits, if any.

3. (10%) Explain the following terms: (1) disk striping (or interleaving) (2) locality.

4. (10%) Explain how to implement "*rendezvous*" interprocess-communication (IPC) mechanism for message-passing systems.

Part II. Compilers (50%)

5. (10%) For the compilation of a C program, which of the following happen or exist at compile time, which at run time, which at both?
(1) type checking (2) bounds checking (3) parameter passing (4) semantic stack
(5) backpatching (6) display (7) lexical analysis (8) parsing (9) symbol table
(10) activation record.

6. (10%) Answer the following true/false questions.
(1) Lex is a lexical analyzer.
(2) In terms of parser size, the SLR and LALR parsing tables for a grammar always have the same number of states.
(3) Some grammars have more shift/reduce conflicts in their LALR parsing table then in their LR(1) parsing table.
(4) The language L = $\{a^nb^n \mid n < 100\}$ is not a regular language.
(5) Transforming a grammar by eliminating all left-recursion and then left-factoring will produce a grammar for which the parsing table has no multiply-defined entries.

7. (10%) In the process of compilation, what are the major kinds of activities done by the semantic actions (rules) associated with the grammar productions.

(背面仍有題目,請繼續作答)

8. (15%) Explain how does a compiler handle procedure calls in program code.

9. (5%) Show the sequence of the productions used by a Yacc parser as it parses the input *id*(id+id)* with respect to the following grammar (note: E is a start symbol):

(1) E -> E + T
(2) E -> T
(3) T -> T * F
(4) T -> F
(5) F -> (E)
(6) F -> id

, where E, T, F are nonterminals and +, *, (, id,) are terminals.