

- [1] 下列為一組表示某種運算式之文法 (grammar)

```

<exp> ::= <term> | <exp> * <term> | <exp> % <term>
<term> ::= <factor> | <term> $ <factor> | <term> # <factor>
<factor> ::= id | ( <exp> )

```

其中 * % \$ # 為運算子 (operators)

請問

- (a) 能否直接根據此文法以 recursive descent 方法設計 parser 以 parsing 以下之運算式？為什麼？(6%)
 $id * id \% id * id \$ id \# id$
- (b) 此文法中那些是 terminals？那些是 nonterminals？(6%)
- (c) 由此文法能否得知運算式之 semantic？為什麼？(6%)
- [2] (a) 試說明一般 (C 或 Pascal) compiler 與 assembler 在其所建的 symbol table 中各記錄那些資料？(12%)
- (b) 在高階語言中，變數之 storage allocation 可分成那幾種方法？(6%)
- [3] 假設有三部計算機 A, B 及 C.
 A 配置有 16M byte real memory, 而其 architecture 最大能提供 64K byte 之 virtual space.
 B 配置有 64K byte real memory, 而其 architecture 最大能提供 16M byte 之 virtual space, 其作業系統不提供 demand paging.
 C 配置有 64K byte real memory, 而其 architecture 最大能提供 16M byte 之 virtual space, 其作業系統提供 demand paging.
 請問在那部(或那幾部)計算機上之 linker 提供 overlay 功能較有意義？為什麼？(10%)
- [4] 試說明用於發展程式之 cross compiler 及 simulator 之功能及其應用場合。(12%)
- [5] 下列為 READER/WRITER processes 對 shared data 作 concurrent access 之 synchronization control.

READER process :

```

P(mutex);
readcount := readcount + 1;
if readcount = 1 then V(wrt);
V(mutex);
...
reading is performed
...

```

```

P(mutex);
readcount := readcount - 1;
if readcount = 1 then P(wrt)
V(mutex);

```

WRITER process :

```

P(wrt);
...
writing is performed
...
V(wrt);

```

以上之 algorithm 能否正確達成 READER/WRITER 之同步要求？若不能，請指出錯誤之處，並將正確版本寫在答案紙上。(10%)

- [6] 以下為 virtual memory system 相關問題。

- (a) 某些 virtual memory operating system 中，在產生 page fault 而 real memory 尚未用完時，(即，尚有 free frame) 即開始作 page replacement, 其原因何在？(5%)
- (b) 在 virtual memory systems 中，那些因素會影響 page fault rate？(請列舉出三種)(9%)
- (c) 一般在 page table entry 中提供 dirty-bit 及 reference-bit 之目的何在？(4%)

[7] 試說明在未提供 virtual memory 之 multiprogramming 系統
(採 round-robin scheduling) 中,

- (a) CPU 之 registers 個數, 作業系統中設定之 time slice 時間
長度, real memory 之大小等因素, 對 response time 之影響。
(8%)
- (b) 在此種系統中, I/O-bound job 是否會有較好(快)之 response
time? 為什麼? (6%)