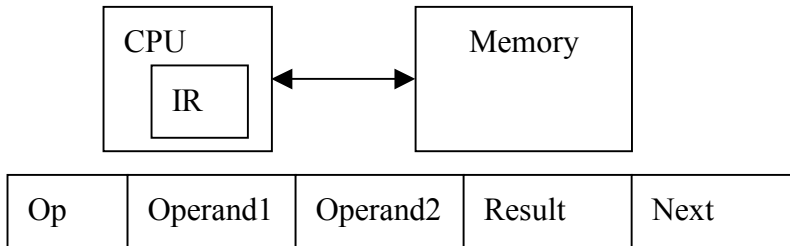


香港中文大學電腦科學系碩士班招生考試試題紙

系 所 組 別	考 試 科 目 (中 文 名 稱)	考 試 日 期	備 註
資 訊 系 乙 組	系統程式	4 月 22 日 第 節	

註：考生可否攜帶計算機或其他資料作答，請在備註欄註明（如未註明，一律不准攜帶）

1. (12%) 某機器 CPU 與 Memory 的結構及指令規格如下：



有一個指令 ADD A, B, C, NEXT. 這個指令將 A 和 B 中資料值相加起來，然後將結果存到符號 C 的位置中，而次一個執行的指令位址值則存在符號 NEXT 中。

- 提供一個方法，以便除去 Next 欄，這需要增加什麼新指令呢？CPU 中要增加一個什麼樣的暫存器(Register)？
- 提供一個方法，以便除去 Result 欄。（不要給 CPU 增加任何的暫存器），這需要那個新指令？
- 另一個除去 Result 欄的方法是在 CPU 中增加一個和 Word 一樣大小的暫存器，稱為累積器 (ACC)，指令運算的結果儲存於累積器中。除了 Result 欄，如果再省略一欄，需增加什麼指令？

2. (12%) 所有的組譯程式(Assembler)做法均相差無幾。若有兩部電腦系統，其指令集 (Instruction Set)雖有所不同，但暫存器的個數與用法均相同。此外，記憶體的字長(Word Length)也均相同。假設在 A 機器上寫一個組譯程式，此組譯程式寫得很好，僅需更換部份資料表即可用來組譯 B 機器的原始程式(Source Program)。在此條件下，試問：

- 二部機器的組合語言應具有何共同特性？
- 那些資料表必須被更換掉，A 機器上原有的組譯程式才能用來在 A 機器上組譯 B 機器上的原始程式。
- 問題(b)所產生的目的程式(Object Program)要如何才能在 A 機器上執行？

3. (10%) 如果一電腦系統使用獨立的繫結程式(Binder)及簡單的載入程式(Loader)。

- 你要從 Direct Linker Loader 中去掉什麼工作（儘量簡化載入程式）？
- 經簡化的載入程式仍需執行什麼工作？
- 此簡化後的載入程式需要幾次處理程序？為什麼？
- 此系統中 O.S. 的 Memory Management 可能是哪種 Strategy？

4. (10%) 請比較 interrupt, cycle stealing 與 DMA(Direct Memory Access)。

5. (10%) 如果巨集處理器(Macro Processor)與組譯程式的 Pass 1 合併處理的話, 請填下表 (複選).

								1. 必須在第一次加以建立或修正
								2. 在第一次被參考到但不加以修正
								3. 在第一次建立以供第二次使用
								4. 在第二次時不加以修正
								5. 在第二次時建立或修正
								6. 在第二次參考到
Symbol Table								
Pseudo Operation Table								
MOT(包含 MNT)								
MDT								
Location Counter								

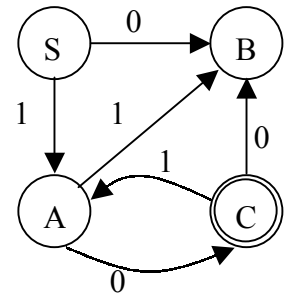
6. (10%) 如果一電腦系統使用獨立的繫結程式(Binder)及簡單的載入程式(Loader).

(e) 你要從 Direct Linker Loader 中去掉什麼工作 (儘量簡化載入程式)?

(f) 經簡化的載入程式仍需執行什麼工作?

(g) 此簡化後的載入程式需要幾次處理程序? 為什麼?

(h) 此系統中 O.S. 的 Memory Management 可能是哪種 Strategy?



7. (10%) 求右圖之有限自動機(Finite Autometa)的

(a) 正規文法(Regular Grammar).

(b) 正規表示法(Regular Expression).

8. (10%) 假設一 page memory system, 其 associative register 的存取速度是 40 nanosecond, 而記憶體存取速度是 800 nanosecond. 如果 90% 的 memory references 可以透過 associative register 搜尋到所要的 page. 請計算出有效的存取時間(effective access time).

9. (16%) 假設一系統包含一個 eraser, 一枝 pencil, 一枝 pen 和兩個 concurrent processes. 呼叫程序 ask(R)會被 block 住直到得到 Resource R 為止.

Process P0	Process P1
(0.1) ask(pencil);	(1.1) ask(pen);
(0.2) use pencil;	(1.2) use pen;
(0.3) ask(eraser);	(1.3) ask(eraser);
(0.4) use pencil and eraser;	(1.4) use pen and eraser;
(0.5) release pencil;	(1.5) release pen;
(0.6) ask(pen);	(1.6) ask(pencil);
(0.7) use pen and eraser;	(1.7) use pencil and eraser;
(0.8) release eraser;	(1.8) release pencil;
(0.9) release pen;	(1.9) release eraser;

(a) 請列出會造成兩個 Processes deadlock 的情況.

(b) 改寫兩個 processes (不改變 use statements 的相對次序) 以避免 deadlock. 請說明你的做法.