

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama  
Sidang Akademik 1999/2000

September 1999

**CTS301/CSC202 - Sistem Pengoperasian**

Masa : [3 jam]

---

**ARAHAN KEPADA CALON:**

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **EMPAT** soalan di dalam **TUJUH** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
  - Anda dibenarkan membawa satu helai kertas A4 yang mengandungi nota sendiri ke dalam bilik peperiksaan.
  - Jawab **SEMUA** soalan dalam Bahasa Malaysia.
-

1. (a) (i) Terangkan perbezaan utama antara sistem interaktif, kelompok (batch) dan masa-nyata (real-time) dan berikan dua keadaan di mana sistem pengoperasian masa-nyata diperlukan. (5/100)

- (ii) Beri penerangan yang ringkas dan tepat bagi panggilan sistem yang digunakan bagi mensimulasi cengkerang Unix (Unix shell):

- fork ()
- execve ()
- wait ()

(5/100)

- (b) Adakah penyelesaian penantian sibuk ("busy waiting") dan pemboleh ubah terkongsi *turn* dapat digunakan apabila dua proses dilarikan pada multipemproses memori-terkongsi (iaitu dua CPU berkongsi memori utama)? Nyatakan kekurangan atau kelebihan (jika ada) kaedah ini.

```
while (TRUE) {
    while (turn != 0);
    critical_region();
    turn = 1;
    noncritical_region();
}
```

Proses 1

```
while (TRUE) {
    while (turn != 1);
    critical_region();
    turn = 0;
    noncritical_region();
}
```

Proses 2

(5/100)

- (c) Diberi atur cara di bawah. Terangkan apa yang akan berlaku sekiranya atur cara tersebut dilaksanakan. Tulis semula keratan kod yang dibaiki.

```
#define N
#define LEFT
#define RIGHT (i-1)%N
#define THINKING
#define HUNGRY
#define EATING
```

```
typedef int semaphore;
int state[N];
semaphore mutex = 1;
semaphore s[N];
```

```
void philosopher(int i)
{
    while (TRUE) {
        think();
        take_forks(i);
        eat();
        put_forks(i);
    }
}
```

```

void take_forks(int i)
{
    down(&mutex);
    state[i] = HUNGRY;
    test(i);
    up(&mutex);
}

void put_forks(int i)
{
    down(&mutex);
    state[i] = THINKING;
    test(LEFT);
    test(RIGHT);
    up(&mutex);
}

void test(i)
{
    if (state[i] == HUNGRY && state[LEFT] != EATING
        && state[RIGHT] != EATING) {
        state[i] = EATING;
        up(&s[i]);
    } else down(&s[i]);
}

```

(10/100)

2. (a) Jelaskan ciri untuk cara-cara perwakilan direktori sesebuah sistem fail yang berikut:

- (i) struktur rata (flat structure)
- (ii) hierarki tegas (strict hierarchy)
- (iii) graf tak berkitar (acyclic graph)

(3/100)

(b) Mengapakah graf berkitar (cyclical graphs) tidak dibenarkan dalam sistem fail? Jelaskan.

(2/100)

(c) Fail sistem Unix membenarkan pengguna melekap (mount) sesuatu peranti pada mana-mana tempat dalam hierarki direktorinya. Namun begitu, peranti yang sama tidak boleh dilekap dua kali pada sistem fail yang sama. Mengapakah terdapat halangan tersebut?

(4/100)

- (d) Fail sistem Unix membekalkan pautan lembut (soft link) dan pautan keras (hard link) pada fail. Pautan lembut mencatatkan maklumat laluan (path information) secara teks kepada unsur direktori fail tertentu, manakala pautan keras mencatatkan *nombor inod* (*inode number*) untuk fail tersebut dalam unsur direktori yang baru.
- (i) Apakah kesan penghapusan unsur direktori yang asal ke atas fail tertentu yang dipautkan secara pautan lembut?
  - (ii) Bolehkah kita capai fail melalui sesuatu pautan keras sekiranya nama unsur direktori yang asal untuk fail tersebut ditukar?
  - (iii) Sekiranya kita perlu mencapai versi (version) terkini fail tertentu, adakah pautan lembut ataupun pautan keras lebih sesuai? Mengapa?
- (6/100)
- (e) Satu sistem fail bersifat eksperimen menggabungkan cara peruntukan blok berdasarkan senarai berpaut dengan cara peruntukan blok berdasarkan peruntukan berdampingan paling cocok (best-fit contiguous block allocation) untuk storan fail pada cakera.

Cakera terdiri daripada blok-blok yang sama saiznya yang mengandungi penuding untuk kegunaan peruntukan senarai berpaut. Tambahan, sistem fail berusaha sebaik mungkin memperuntukkan satu julat blok yang berdampingan (range of consecutive blocks) secara senarai berpaut, melalui algoritma peruntukan paling cocok.

Namun, sekiranya saiz fail tersebut melebihi saiz blok berdampingan yang terbesar, maka fail tersebut akan diperuntukan mengikut beberapa bahagian dengan menggunakan kaedah senarai berpaut berdasarkan algoritma berikut:

- (i) Julat blok bebas berdampingan yang terbesar dipilih untuk menyimpan data dalam fail yang belum diproses.
- (ii) Sekiranya data yang tertinggal masih melebihi saiz blok bebas berdampingan yang terbesar, langkah (i) diulangi.
- (iii) Jika tidak, data yang tertinggal disimpan secara algoritma peruntukan paling cocok.
- (iv) Blok yang tertinggal dalam siri berdampingan tersebut boleh digunakan untuk peruntukan seterusnya.

Contoh, blok bebas berdampingan bersaiz 12K, 16K, 8K boleh digunakan. Fail bersaiz 20K harus disimpan.

Maka fail tersebut akan disimpan sebagai 16K+4K menggunakan teknik yang dijelaskan, dan blok bebas berdampingan yang tertinggal mempunyai saiz 12K dan 4K selepas pemrosesan fail tersebut.

Diberi bahawa setiap blok bersaiz 4K, dan fail-fail berikut akan dicipta secara berjuran dalam sistem fail tersebut:

fail\_a (12K), fail\_b (36K), fail\_c (4K), fail\_d (20K), fail\_e (8K). Satu peta bit ruang bebas (free space bitmap) untuk cakera tersebut diberi seterusnya (0=blok bebas, 1=blok yang digunakan, Nombor blok bermula dengan 0):

Nombor Blok	0.....7	8.....15	16.....23	24.....31
Peta bit ruang bebas	1 1 0 0 0 0 0 1	1 1 1 0 0 0 0 1	1 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0

Berikan senarai blok yang akan diperuntukkan untuk setiap fail berdasarkan algoritma yang diberi tersebut.

(10/100)

3. (a) Terangkan perbezaan antara serpihan dalaman (internal fragmentation) dan serpihan luaran (external fragmentation). Yang manakah antara kedua-duanya berlaku pada sistem penghalaman dan yang mana berlaku apabila menggunakan pengsemanan tulen?

(4/100)

- (b) Apakah yang dimaksudkan dengan panggilan prosedur jauh (RPC) dan terangkan bilakah pemetakan data (data partitioning) lebih sesuai digunakan berbanding pemetakan fungsi (function partitioning) dan sebaliknya.

(5/100)

- (c) Suatu komputer mengandungi empat kerangka. Pada detik jam pertama, bit R (bit Referenced) adalah 0111 (halaman 0 ialah 0 dan lainnya 1). Pada detik jam berikutnya nilai bagi R ialah 0010, 1010, 1100, 1011, 1010, 1101 dan 0001. Jika algoritma ketuaan (aging algorithm) yang menggunakan pengira 8 bit. Berikan nilai keempat-empat pengira selepas detik terakhir.

(4/100)

- (d) Suatu komputer mengandungi empat kerangka halaman. Masa muatan (loading), masa capaian terakhir, dan masa nilai bit R (Referenced) dan M (Modified) bagi setiap halaman adalah seperti di bawah (masa diberikan dalam detik jam):

Halaman	Muatan	Rujukan Terakhir	R	M
0	121	274	0	1
1	225	255	1	0
2	115	267	1	1
3	155	275	1	1

- (i) Halaman mana yang akan digantikan oleh "bukan terkemudian digunakan" (NRU)?
- (ii) Halaman mana yang akan digantikan oleh "masuk-dulu keluar-dulu" (FIFO)?
- (iii) Halaman mana yang akan digantikan oleh "paling lama tidak digunakan" (LRU)?
- (iv) Halaman mana yang akan digantikan oleh "peluang kedua" (second chance)?

Terangkan jawapan yang anda berikan.

(12/100)

4. (a) Satu matriks pencapaian (access matrix) ditetapkan untuk pencapaian sumber dalam sistem tertentu. Subjek  $S_1$ ,  $S_2$ , serta fail  $F_1$ ,  $F_2$  dan peranti  $D_1$  terdapat dalam sistem tersebut.

Matriks Pencapaian	$S_1$	$S_2$	$F_1$	$F_2$	$D_1$
$S_1$	Kawal (control)		Laku (execute) Hapus (delete)	Baca (read) Tulis (write)	
$S_2$	Pemilik (owner)	Kawal (control)		Baca (read)	Jangkau (seek) Pemilik (owner)

Untuk setiap kes berikut, adakah pencapaian sumber tersebut dibenarkan? (Sila jawab Ya atau Tidak)

- (i)  $S_1$  baca  $F_2$
  - (ii)  $S_2$  memiliki  $S_1$
  - (iii)  $S_1$  hapus  $F_2$
  - (iv)  $S_2$  baca  $F_2$
  - (v)  $S_1$  jangkau  $D_1$
- (b) Protokol Pengesahan Rangkaian Kerberos membekalkan tiket kepada pelanggan untuk tujuan pencapaian pelayan.
- (i) Apakah kandungan tiket tersebut?
  - (ii) Jelaskan langkah yang berlaku dalam pelayan selepas pelanggan menghantar tiket tersebut kepada pelayan sehingga perkhidmatan pelayan dimulakan.
  - (iii) Jika sesuatu mesin penyamar memperoleh tiket tersebut, mengapakah ia tidak dapat menyamar sebagai pelanggan yang sebenar?

(5/100)

(5/100)

- (c) Secara menggunakan algoritma Pengurus Bank (Banker's algorithm), tentukan sama ada peruntukan sumber (resource allocation) sistem berikut selamat atau tidak selamat (safe or unsafe). Tunjukkan langkah penyelesaian anda.

Jenis sumber  $R_0, R_1, R_2$  dan  $R_3$  terdapat dalam sistem dengan bilangan

$$C = \langle 3, 3, 2, 4 \rangle.$$

Proses dilakukan secara selari (simultaneously) dalam sistem ialah:

$P_0, P_1, P_2$ , dan  $P_3$ .

Tuntutan maksimum (maximum claims) untuk setiap proses adalah seperti berikut:

Tuntutan Maksimum	$R_0$	$R_1$	$R_2$	$R_3$
$P_0$	2	0	1	0
$P_1$	2	1	2	3
$P_2$	3	1	1	1
$P_3$	0	0	2	4

Kedaaan sistem terkini ialah:

Kedaaan terkini	$R_0$	$R_1$	$R_2$	$R_3$
$P_0$	1	0	1	0
$P_1$	0	1	0	1
$P_2$	0	1	1	0
$P_3$	0	0	0	1

(10/100)

- (d) Suatu proses baru  $P_4$  yang mempunyai tuntutan maksimum  $\langle 3, 2, 0, 1 \rangle$  dimulakan sebaik sahaja proses pertama dalam bahagian (c) tamat, dan berjaya memperoleh sumber  $\langle 0, 0, 0, 1 \rangle$  sebelum pelaksanaan proses seterusnya. Adakah sistem tersebut masih selamat? Mengapa?

(5/100)