

UNIVERSITI SAINS MALAYSIA

Peperiksaan Tambahan
Sidang Akademik 1996/97

Mei 1997

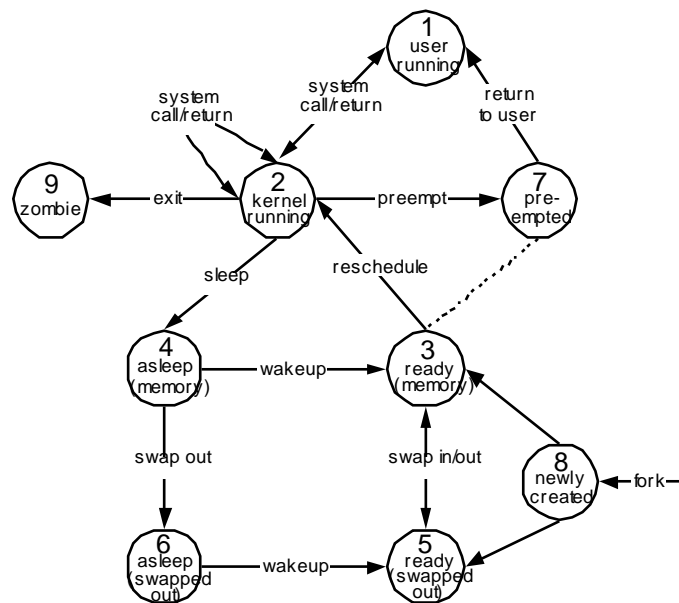
CSC202 - Sistem Pengoperasian

Masa : [3 jam]

ARAHAN KEPADA CALON:

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **LAPAN** soalan di dalam **LAPAN** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
 - Jawab **ENAM** daripada LAPAN soalan. Jawab soalan dalam Bahasa Malaysia.
-

1.



Gambarajah peralihan keadaan di atas melambangkan model proses Unix.

- (a) Katakan sesuatu proses **P** berada dalam keadaan **1** (*pengguna-laksana*). Kenapakah proses beralih ke keadaan **2** (*kernel-laksana*) sebelum sesuatu panggilan IO boleh diberikan perkhidmatan? (20/100)
- (b) Katakan panggilan IO tersebut memerlukan sumber sistem yang belum sedia digunakan. Jelaskan apa yang akan berlaku dalam konteks model proses di atas. Apakah kelebihan menempatkan proses P dalam keadaan selain daripada 1, 2 ataupun 3? (20/100)
- (c) Bezakan keadaan **4** (*tidur-ingatan*) dan **6** (*tidur-terswap*). Bilakah peralihan **4** \leftrightarrow **6** biasanya berlaku? Apakah kelebihan sistem pengoperasian dengan keupayaan swap-keluar? (20/100)
- (d) Berikan justifikasi bagi peralihan dari keadaan **3** (*sedia-ingatan*) \times **5** (*sedia-terswap*), dan apakah sebabnya peralihan **6** \leftrightarrow **4** tidak diimplementasikan dalam Unix? (20/100)
- (e) Bezakan proses dalam keadaan **5** dengan proses dalam sesuatu sistem bermemori maya. Apakah kelebihan keadaan terswap (**5** dan **6**) dalam konteks sistem bermemori maya?

2. (a) Dengan menggunakan pemetaan (partitioning) dinamik, tunjukkan keadaan sesuatu memori (saiz 800 Kbyte) untuk jujukan operasi berikut:

Proses 1 masuk
 Proses 2 masuk
 Proses 3 masuk
 Proses 2 keluar
 Proses 4 masuk
 Proses 3 keluar
 Proses 5 masuk
 Proses 6 masuk

Saiz proses-proses adalah seperti berikut:

<i>Proses</i>	<i>Saiz (Kbyte)</i>
P1	50
P2	100
P3	350
P4	200
P5	100
P6	150

(35/100)

- (b) Sesuatu memori dibahagikan kepada 12 kerangka (frame). Tunjukkan keadaan memori bagi jujukan operasi seperti di atas. Saiz proses-proses adalah seperti berikut:

<i>Proses</i>	<i>Saiz (halaman)</i>
P1	2
P2	3
P3	2
P4	5
P5	1
P6	4

(35/100)

- (c) Lakarkan keadaan memori dan apabila berlakunya kesilapan halaman (page fault) bagi polisi gantian *Least Recently Used* (LRU). Saiz memori adalah 4 halaman, dan jujukan alamat halaman adalah seperti berikut:

2 4 5 2 1 4 5 3 4 2 1 6 3 6 1 2

3. (a) Jadual berikut menunjukkan masa ketibaan dan masa pelaksanaan beberapa proses yang perlu dilaksanakan oleh suatu sistem pengoperasian berbilang-tugas.

<i>ID Proses</i>	<i>Masa ketibaan (ms)</i>	<i>Masa pelaksanaan (ms)</i>
1	0	4
2	1	6
3	3	2
4	5	5
5	6	1
6	8	4

Tunjukkan pelaksanaan (dan juga keadaan semua queue yang berkaitan) proses-proses di atas mengikut polisi-polisi penjadualan berikut:

- (i) "First Come First Served"
- (ii) "Shortest Process Next"
- (iii) "Shortest Remaining Time"

(50/100)

- (b) Jadual berikut menunjukkan jujukan permintaan bagi sesuatu sistem cakera dengan silindernya dinomborkan dari 0 (paling dalam) ke 99.

<i>Masa ketibaan (ms)</i>	<i>Permintaan silinder</i>
1	24
3	32
7	75
11	63
12	76
19	42
22	10
26	50
30	82
34	43
35	70
39	33

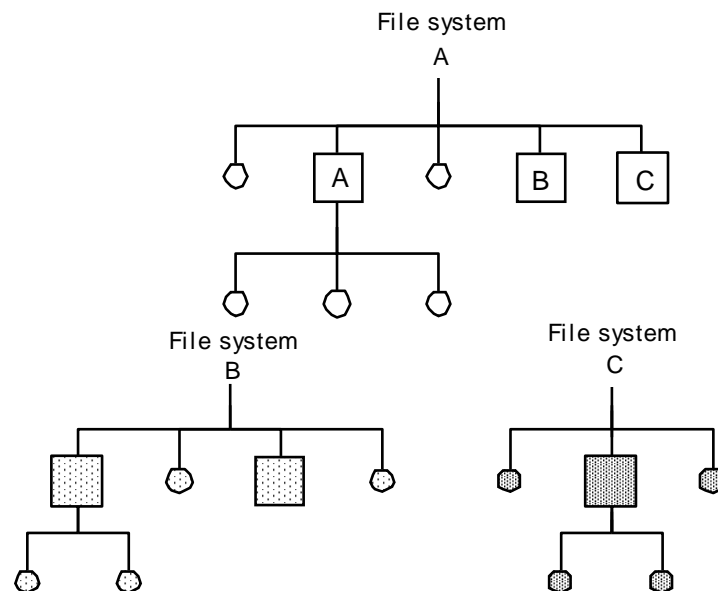
Kepala baca-tulis bermula dari silinder 0, dan pada mulanya bergerak ke arah pinggir cakera. Berikan jujukan perkhidmatan bagi polisi-polisi penjadualan berikut:

- (i) Elevator bersegmen (dengan saiz segmen penimbal = 4)
- (ii) Shortest Service Next bersegmen (dengan saiz segmen penimbal = 6)

(iii) Shortest Service Next bersegmen (dengan peralihan penimbal setiap 10 ms)

(50/100)

4.



- (a) Jelaskan operasi "*mount*" dalam konteks Unix. Lakarkan sistem fail logikal yang terbentuk bila sistem fail **B** dan **C** di"*mount*"kan ke atas direktori **B** dan **C** dalam sistem fail akar **A**. (25/100)
- (b) Kedudukan fizikal sistem fail **A** dan **B** adalah dalam cakera yang sama. Jelaskan konsep sistem pengurusan fail yang membolehkan dua sistem fail berasingan wujud dalam cakera fizikal yang sama. Apakah kelebihan utama skema sebegini? (25/100)
- (c) Model proses Unix mengandungi dua keadaan swap-keluar iaitu *sedia-terswap* dan *tidur-terswap*). Mana antara sistem fail di atas sepatutnya mengandungi imej proses yang telah terswap keluar kepada memori sekunder? (25/100)
- (d) Dengan menggunakan gambarajah di atas, jelaskan bagaimana sesuatu fail di sistem **B** boleh dipautkan kepada satu fail di sistem **C**. Apakah bezanya antara salinan dengan pautan dalam konteks sistem pengurusan direktori? (25/100)

5. (a) Jelaskan secara ringkas dua pendekatan untuk melindungi fail laluan-kata. (20/100)

(b) Bagi Matriks Capaian berikut, tunjukkan:

- (i) Senarai Kawalan Capaian
(ii) Tiket Keupayaan

	Program 1	Program 2	Program 3	Segmen A	Segmen B
Proses 1	R	R W E	W	R	R W
Proses 2		R W E	R		R W E
Proses 3	R W		R W E	R W	R W

R = Baca

W = Tulis

E = Laksana

(40/100)

- (c) Tuliskan satu atur cara mudah untuk menunjukkan struktur sesuatu virus. (40/100)

6. (a) Jelaskan konsep senibina pelanggan-pelayan. Tunjukkan bagaimana senibina tersebut membolehkan interaksi antara komponen di stesen-kerja pelanggan dan pelayan. (25/100)

- (b) Apakah fungsi asas senibina OSI? Lakar dan labelkan semua lapisan OSI. Berikan huraian ringkas bagi setiap lapisan. (25/100)

- (c) Lakar dan labelkan lapisan senibina TCP/IP. Berikan huraian ringkas bagi setiap lapisan, dan bandingkan dengan senibina OSI. (25/100)

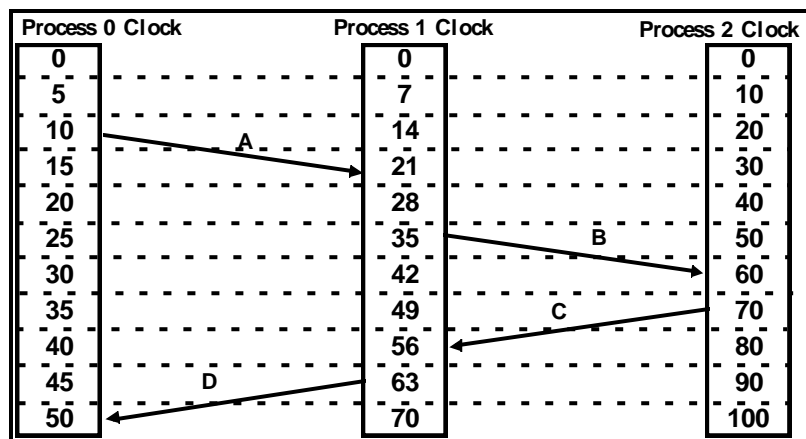
(d) Berikan definisi Sistem Teragih, dan jelaskan konsepnya. Dengan ringkas, huraikan kelebihan sesuatu sistem teragih berbanding dengan PC terasing.

(25/100)

7. (a) Senaraikan tiga cadangan Lamport untuk penyegerakan jam dalam sistem teragih.

(30/100)

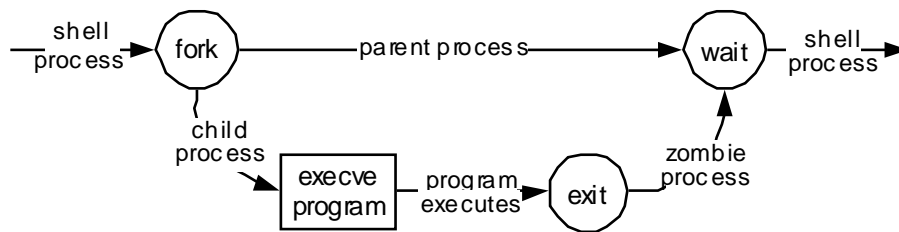
- (b) Dalam sesuatu sistem teragih, andaikan tiga proses yang berlaksana atas mesin berasingan. Setiap proses mempunyai jam tersendiri seperti yang ditunjukkan di bawah.



Gunakan *Algoritma Lamport* untuk memberikan masa kepada kejadian di atas supaya ketiga-tiga proses adalah segerak. Anda hanya perlu menunjukkan jam terkemaskini bagi setiap proses.

(70/100)

8.



Gambarajah di atas menghuraikan jujukan kejadian disebabkan input baris-arahan Unix berikut:

$\$ \text{program } arg1 \text{ } arg2 \dots argn$

di mana cangkerang (antaramuka-pengguna) menggunakan panggilan sistem *fork* untuk melaksanakan boleh-laksana *program* sebagai proses berasingan.

- (a) Terdapat beberapa arahan *dalam* yang tidak di "*fork*"kan, tetapi dilaksanakan dalam ruang alamat proses cankerang. Contohnya arahan *exit* untuk menamatkan sesi kongsi-masa. Kenapakah *exit* tidak patut diimplementasikan sebagai proses berasingan?
(25/100)
- (b) Apakah yang dilakukan oleh cangkerang Unix jika senarai argumen program untuk dilaksanakan mengandungi aksara "*wild-card*"?
(25/100)
- (c) Selepas tamat pelaksanaan, proses anak membentuk suatu proses "zombie". Bagaimanakah keadaan ini boleh digunakan oleh proses ibu (cangkerang)?
(25/100)
- (d) Apakah yang berlaku kepada proses anak bila cangkerang tamat secara tidak normal?
(25/100)