

## PENJADWALAN PROSES (PROCESS SCHEDULING)

## 1) First Come First served (non-preemptive)

- Proses dikirim berdasarkan waktu kedatangan dalam antrian.
- Sekali CPU menjalankan suatu proses, akan dijalankan hingga selesai.
- Menggunakan antrian FIFO, pekerjaan pertama dipilih dan antrian pertama dan diselesaikan hingga CPU burst komplit.

• Kelebihan : Mudah diprediksi, lebih sederhana dan low overhead.

• Kelemahan : tidak dapat menjamin waktu respon yang baik, waktu respon rata-rata cukup panjang, tidak tepat digunakan dalam sistem yang interaktif dan memungkinkan terjadinya fluktuasi yang besar dalam average turn-around time.

## 2) Shortes Job First (preemptive &amp; non-preemptive)

- Mengambil proses dengan burst time paling kecil untuk dijalankan terlebih dahulu.
- Tujuan : mengoptimisasi average waiting time suatu proses.

• Kelebihan : tidak selalu meminimalkan average turn-around time, dapat menimbulkan starvation, tidak bisa memprediksi CPU burst time untuk proses selanjutnya.

## 3) Priority Scheduling (Preemptive &amp; non-preemptive)

- alokasi proses CPU berdasarkan prioritas tertinggi, jika sama maka menggunakan FCFS.

## • Prioritas Skala :

► Internal : batas waktu, memori yang dibutuhkan, jumlah file yang terbuka, perbandingan rata-rata waktu IO dan CPU.

► Eksternal : penggunaan rata-rata pemakaian komputer.

• Kelembahan : starvation yang terjadi pada proses dengan prioritas rendah, maka dari itu dilakukan aging sejalan dengan berjalananya proses tersebut maka prioritas proses juga akan meningkat.

## 4) Round Robin (preemptive)

- Tidak memiliki prioritas, & algoritma paling sederhana.
- proses dikirim secara FIFO tetapi dengan waktu CPU terbatas (time slice / kuantum). Jika proses belum selesai sebelum waktu CPU habis, maka CPU akan di preempted dan diberikan kepada proses selanjutnya yang sedang menunggu di queue. proses yang di preempted kemudian diletakkan di bagian belakang ready list.

## • Permasalahan : Panjang kuantum

- jika terlalu pendek akan menyebabkan terlalu banyak switch konteks dan menurunkan efisiensi CPU.
- jika terlalu panjang akan menyebabkan waktu respons yang lama dan mendekati FCFS.

► Waiting time : waktu terhitung sejak proses tersebut masuk ke dalam antrian sampai proses tersebut mulai dijalankan.

► Turnaround time : waktu dari sejak proses dimasukkan ke dalam antrian (dari waktu menunggu) sampai proses tersebut selesai dijalankan.

► Pre-emptive : suatu sistem dapat di-interrupt di tengah-tengah ketika proses sedang berjalan. proses tersebut akan dihentikan kembali ketika proses lain yang menginterruptnya sudah selesai menggunakan prosesor.

- Non-preemptive : sistem dimana ketika ada proses yang melakukan sedang berjalan, proses tersebut tidak dapat diinterupsi sampai proses tersebut selesai dilayani.
- Quantum time pada Round Robin berguna untuk membatasi proses sehingga proses lain mendapatkan "jatah".
- Performa Round Robin bergantung pada besar Quantum time
- Perbedaan Multilevel Feedback Queue algorithm dengan Multilevel Queue Algorithm → MFC mengizinkan proses untuk pindah antar antrian, MQ tidak mengizinkan.
- Parameter MFC's : (1) jumlah queue yang digunakan. (2) algoritma penjadwalan untuk setiap queue (3) Metode yang digunakan untuk menentukan kapan prioritas sebuah queue ditingkatkan. (4) Metode yang digunakan untuk menentukan kapan prioritas sebuah queue diturunkan. (5) Metode yg digunakan untuk menentukan di queue mana suatu proses seharusnya masuk untuk dilayani.

## STATUS PROSES.

- New : status yang dimiliki pada saat proses baru saja dibuat
  - Run : proses sedang dijalankan (di eksekusi oleh prosesor)
  - Wait : proses sedang menunggu untuk dimasukkan dalam tahap ready.
  - Ready : proses siap untuk dijalankan prosesor.
  - Terminated : status saat proses telah selesai di eksekusi
  - Utilitas CPU : hasil bagi antara jumlah run dibagi dgn time unit.
  - Average Load : hasil bagi antara jumlah semua state yang ready bagi satuan waktu, dengan jumlah maks satuan waktunya.
  - Multiprogramming degree : banyaknya proses yang sedang berjalan di CPU.
  - CPU Busy : persentase penggunaan CPU untuk semua proses.
  - Arrival time : kapan proses itu datang, sehingga kita bisa mengetahui proses mana saja yang sedang berjalan pada suatu waktu.
  - CPU Time : waktu total suatu proses sampai habis.
- Nama / keterangan (event) penyebab perubahan status proses:
- Admitted : proses yang baru dicalonkan akan mempunyai ready state.
  - Scheduler dispatch : proses ber state ready menjadi running, karena penjadwalan memutuskan penggunaan pemrosesan untuk proses itu karena proses yang saat itu berada state-nya (menjadi running atau waiting) atau telah selesai sehingga disingkirkan dari sistem. proses menjadikannya mendapatkan jatah pemrosesan.
  - I/O or event wait : sebuah proses ber state running menjadi waiting karena proses tersebut menunggu suatu kejadian tertentu misal selesainya operasi I/O. Apabila kejadian yang ditunggu telah selesai maka proses dipindahkan kembali ke antrian ready dan siap dijalankan.
  - Interrupt : sebuah proses dari keadaan running dapat menjadi ready kembali karena diinterupsi oleh proses lain. Interupsi dapat disebabkan karena jatah waktu yang diberikan CPU ke proses telah habis sementara proses masih memerlukan sejumlah waktu untuk selesai.
  - I/O or event completion : proses ber state waiting menjadi ready saat sumbu daya yang diminta/diperlukan telah tersedia / tayangan operasi I/O selesai (event occurs).
  - Exit : sebuah state ber state running menjadi terminated disebabkan
    - 1) proses memang sdh selesai, sehingga diakhiri secara normal
    - 2) Melalui batas waktu yang diberikan
    - 3) Terjadi kesalahan perhitungan, division by zero.
    - 4) Terjadi kegagalan I/O, misal kegagalan pembacaan / penulisannya.
    - 5) proses induknya berakhir.
    - 6) proses child diakhiri atas permintaan proses parent-nya Parent kirim signal, SIGQUIT, SIGKILL atau SIGTERM.

<p>④ Sistem Operasi.</p> <ul style="list-style-type: none"> <li>- Berfungsi sebagai penengah antara user dan hardware.</li> </ul> <p>► Goals : eksekusi program &amp; solving problems lebih mudah, membuat sistem komputer mudah digunakan, penggunaan hardware komputer lebih efisien.</p> <p>► Struktur : Hardware (CPU, memory, I/O devices), OS, program aplikasi, users</p> <p>► Peran :       <ul style="list-style-type: none"> <li>- resource allocator</li> <li>- Control program.</li> </ul> <ul style="list-style-type: none"> <li>- kernel : basic program yang berjalan</li> <li>- Trap / exception : software-generated interrupt yg disebabkan oleh error atau request dari user.</li> <li>- An OS is interrupt driven.</li> <li>- Main memory : random access, volatile</li> <li>- Solid-state disks : lebih cepat dan hard-disk, nonvolatile</li> <li>- Caching : mengcopy informasi ke storage system yang lebih cepat.</li> <li>- Device driver untuk setiap device controller manage I/O.</li> </ul> <p style="margin-left: 20px;">main memory → cache → registers.</p> </p> <p>► Multiprocessors : increased throughput, economy of scale, increased reliability.</p> <ul style="list-style-type: none"> <li>- Asymmetric : 1 processor untuk 1 task</li> <li>- Symmetric : 1 processor untuk banyak task.</li> <li>- Multi-chip &amp; multi-core : lebih cepat.</li> </ul> <p>► Clustered Systems : multiple systems bekerja bersama-sama. via SAN. Provides high-availability &amp; high performance computing.</p> <ul style="list-style-type: none"> <li>- Asymmetric : 1 mesin hot-standby mode</li> <li>- symmetric : multiple nodes monitoring each other</li> <li>- Distributed lock manager untuk menghindari konflik.</li> </ul> <p>► Multi programming : organizes jobs, pasti selalu ada 1 yang dieksekusi CPU. Dijalankan dengan job scheduling. (Batch System)</p> <p>► Multitasking (timesharing) : menjalankan banyak task secara teratur. Dijalankan dengan CPU scheduling. response time &lt; 1 s</p> <p>► Hardware interrupt : kirim ke sebuah device</p> <p>► Software interrupt &amp; software error (division by zero), request for operating system service, infinite loop.</p> <p>► Mode       <ul style="list-style-type: none"> <li>→ user mode (bit 1) terkena interrupt</li> <li>→ kernel mode (bit 0)</li> </ul> </p> <p>► Process management : proses adalah program yang dieksekusi. Program (pasif), proses (aktif).</p> <ul style="list-style-type: none"> <li>- Single-threaded : 1 program counter.</li> <li>- Multi-threaded : 1 program counter / thread.</li> </ul> <p>► creating, deleting, suspending &amp; resuming, menyediakan resources untuk sinkronisasi, komunikasi proses dan deadlock handling.</p> <p>► Memory management : Untuk mengeksekusi program, instruksi harus ada di memori.</p> <p>► alokasi memori, memanage bagian dari memori yg digunakan oleh siapa, memenuhi proses keluar/masuk memori.</p>	<p>► File storage management : organized into directories. Access control to determine who can access what.</p> <p>► creating &amp; deleting files, manipulation, mapping files onto secondary storage, Backup Files</p> <p>► Mass storage management : disks digunakan untuk menyimpan data yg harus disimpan dalam jangka waktu yang lama.</p> <p>► Free space management, storage allocation, disk scheduling.</p> <p>► I/O subsystem : buffering (storing data while being transferred), caching, queuing (menyajikan output Job 1, input Job 2).</p> <p>► Protection : mengontrol akses pada program proses, user pada sistem, resource dr user.</p> <p>► Security : defense of the system against internal and external attacks.</p> <p>► Computing Environment :       <ul style="list-style-type: none"> <li>- Traditional : general purpose machine</li> <li>- Mobile : handheld smartphones, allows augmented reality.</li> <li>- Distributed : collection of separate, heterogenous systems networked together LAN, MAN, WAN, PAN</li> <li>- Client server : compute (client kirim request service), file (client menyimpan &amp; menerima file).</li> <li>- Peer-to-peer : tidak ada client server skype, (voice over IP).</li> <li>- Virtualization : emulation &amp; virtualization</li> <li>- Cloud computing : public, private, hybrid.</li> <li>- real time embedded system : microcontroller</li> </ul> </p> <p>► HAKI</p> <p>merupakan hak-hak (wawancara/ketulusan) untuk berbuat sesuatu atas keleluasaan intelektual, yang diatur oleh norma-norma atau hukum-hukum yang berlaku.</p> <p>► Hak Cipta : Hak eksklusif bagi pencipta atau penerima hak untuk mengumumkan atau memperbaiki ciptaan nya atau memberikan izin untuk itu dengan tidak mengurangi pembatasan menurut peraturan perundang-undangan yang berlaku.</p> <p>► Hak Paten : Hak eksklusif yang diberikan oleh negara kepada investor atas hasil inovasinya di bidang teknologi, yang untuk selama waktu tertentu melawan sendiri inovasinya tersebut atau memberikan persetujuannya kepada pihak lain untuk melaksanakannya.</p> <p>► Merk dagang (trademark) : tanda yang berupa nama, gambar, kata, huruf, angka, susunan warna atau kombinasi dari unsur-unsur tersebut yang memiliki daya pembeda dan digunakan dalam kegiatan perdagangan barang/jasa. cc : Kacang Atom cap ayam santan.</p> <p>► Rahasia dagang (trade secret) : informasi yang tidak diketahui oleh umum di bidang teknologi / bisnis, mempunyai nilai ekonomi karena berguna dalam kegiatan usaha, dan dijaga kerahasiannya oleh pemilik. Rahasia dagang. co : rahasia formula R&amp;R</p> <p>► Service mark : kata, frase, logo, simbol, warna, bali yang digunakan oleh sebuah bisnis untuk mengidentifikasi sebuah layanan dan membedakannya dari kompetitornya. cc : Pegadaran : mengatasi masalah tanpa masalah.</p>	<p>► Perangkat Lunak Bebas : perangkat lunak yg mengizinkan siapapun untuk menggunakan, menyulut, mendistribusikan, baik dimodifikasi atau pun tidak, secara gratis ataupun dengan biaya.</p> <ul style="list-style-type: none"> <li>- Kebebasan 0 : bebas u/ menjalankan program u/ tujuan apapun saja.</li> <li>- Kebebasan 1 : bebas u/ mempelajari bagaimana program itu berjalan serta dapat digesuaikan dengan kebutuhan.</li> <li>- Kebebasan 2 : bebas u/ mengkaji-kaji dan membaca hasil sajinan perangkat lunak tersebut.</li> <li>- Kebebasan 3 : bebas u/ meningkatkan kinerja program &amp; menjelaskannya.</li> <li>- Kebebasan 4 : bebas u/ memodifikasi perangkat lunak tersebut.</li> </ul> <p>► Perangkat Lunak Bebas / Free software bukan berarti tidak komersial.</p> <p>► Copyleft : merupakan PLB yg turunannya tetap PLB. contoh lisensi copyleft adalah GNU / GPL, General Public License. Perangkat lunak bebas yg ketentuan pendistribusianya tidak memperbolehkan untuk menambahkan batasan-batasan tambahan - jika mendistribusikan atau memodifikasi perangkat lunak tersebut.</p> <p>► PLB non-copyleft dibuat, mengizinkan seseorang untuk mendistribusikannya dan memodifikasinya dan untuk menambahkan batasan-batasan tambahan didalamnya.</p> <p>► Jika suatu PL bebas tetapi tidak copy-left, beberapa salinan / versi yg dimodifikasi bisa jadi tidak bebas.</p> <p>► Contoh Lisensi : Apache, Artistic, Free BSD, openLDAP, dll.</p> <p>► Tantangan : perangkat keras rahasia, pusat data bebas, paten perangkat lunak, Dokumentasi bebas.</p>
---	---	---

## PROTECTION AND SECURITY

- Proteksi adalah mekanisme sistem-operasi untuk mengontrol akses terhadap beberapa objek yg dilindungi / dijaga dalam sistem operasi. objek-objek tsb bisa berupa perangkat keras/lunak. Mekanisme proteksi dapat digambarkan dengan konsep domain. Domain adalah himpunan yg bersifat pasangan objek dan hak akses. Proses bisa berpindah dari domain ke domain lain dalam aksesnya. Bekerja secara pasif. Setiap proses diberikan akses yg cukup, tidak lebih dari resource yg ditunjuk.
- Security (keamanan) adalah melindungi data-data dan informasi berharga dari pihak-pihak yg tidak berhak. Bekerja secara aktif. Sistem komputer dan data-data di dalamnya terencam dari aspek :
- 1.) Intruders : penyusup (attempt to break security)
  - 2.) Threats : Ancaman (potential security violation)
  - 3.) Attack : attempt to break security.

## 13 ADWALAN PROSES.

### 1) First Come First served.

\* Soal 1.

$P_1(10), P_2(1), P_3(2), P_4(1), P_5(5)$ .

$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
10	11	13	14	19

Average waiting time :

$$(0+10+11+13+14)/5 = 9.6 \text{ s}$$

Average turn-around time :

$$(10+11+13+14+19)/5 = 13.4 \text{ s}$$

WATING	TURNAROUND
$T_1=0$	$T_1=10$
$P_1=0.5$	$P_1=10.5$
$P_2=10.5$	$P_2=11.5$
$P_3=11.5$	$P_3=13.5$
$P_4=13.5$	$P_4=14.5$
$P_5=14.5$	$P_5=19.5$

Average turn-around time :

$$(10+11+13+14+19)/5 = 13.4 \text{ s}$$

### 2) Shortest Job First

\* Soal 1.

$P_1(12), P_2(15), P_3(19), P_4(2), P_5(7), P_6(9), P_7(16)$ .

$P_4$	$P_5$	$P_6$	$P_1$	$P_2$	$P_7$	$P_3$
2	9	18	30	45	61	80

1) Waiting time top proses

$$P_1=18 \text{ s} ; P_2=30 \text{ s} ; P_3=61 \text{ s} ; P_4=0 \text{ s} ; P_5=2 \text{ s} ; P_6=9 \text{ s} ; P_7=45 \text{ s}$$

### 3.) Priority scheduling

\* Soal 1 : Process (Burst time; Arrival time; Priority)

$P_1(10:0:3), P_2(1:0:1), P_3(2:0:4), P_4(1:0:5)$

$P_5(5:0:2)$ .

$P_2$	$P_5$	$P_1$	$P_3$	$P_4$
1	6	10	16	18

Average waiting time :  $(0+1+6+10+16+18)/5 = 8.2$

Average response time :  $(0+1+6+10+16+18)/5 = 8.2$

Average turn-around time :  $(10+16+18+10+16+18)/5 = 14$

worst case : starvation, blocking of low priority processes

Aging : keterbatasan dalam proses prioritas rendah.

teknik bertahap meningkatkan prioritas dan proses yang menunggu lama dalam sistem untuk jangka waktu yg lama.

### 4) Round Robin Scheduling.

\* Soal 1 (Penjadwalan Proses 2 (2002)).

5 proses tiba secara bersamaan pada saat  $t=0$  dengan urutan  $P_1, P_2, P_3, P_4$  dan  $P_5$ . Bandingkan waiting time & turnaround time dari kelima proses jika mengimplementasi RR dengan quantum 2 satuan waktu.

a) Burst time kelima proses secara berturut-turut :

$$(10, 8, 6, 4, 2) \text{ satuan waktu.}$$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
$P_1$	$P_{10}$	$P_9$	$P_{11}$	$P_{12}$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$R_2$	$R_7$	$N_9$	$W_{10}$	$W_{11}$
$P_2$	$W_1$	$W_2$	$R_2$	$R_7$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$	$W_{10}$	$R_6$	$R_5$	$W_{11}$
$P_3$	$W_1$	$W_2$	$W_3$	$W_4$	$P_6$	$R_5$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$	$W_{10}$	$W_{11}$	$W_{12}$	$R_4$
$P_4$	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$R_4$	$P_3$	$W_7$	$W_8$	$W_9$	$W_{10}$	$W_{11}$	$W_{12}$	$W_{13}$
$P_5$	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$R_2$	$R_1$	—	—	—	—	—

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$P_1$	$W_{12}$	$W_{13}, W_{14}$	$R_6$	$R_5$	$W_{15}, W_{16}, W_{17}$	$W_{18}, W_{19}$	$R_4$	$P_3$	$W_{19}, W_{20}$	$R_2$	$R_1$	—	—	—
$P_2$	$W_{12}$	$W_{13}, W_{14}$	$W_{15}$	$W_{16}$	$R_4$	$R_3$	$W_{17}, W_{18}$	$W_{19}, W_{20}$	$R_2$	$R_1$	—	—	—	—
$P_3$	$P_5$	$W_3$	$W_4$	$W_{15}, W_{16}$	$W_{17}$	$W_{18}$	$R_2$	$R_1$	—	—	—	—	—	—
$P_4$	$W_{14}$	$R_2$	$R_1$	—	—	—	—	—	—	—	—	—	—	—
$P_5$	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Turnaround time : Waiting time :

$$P_1=30 \text{ s} ; P_5=10 \text{ s}$$

$$P_2=28 \text{ s}$$

$$P_3=24 \text{ s}$$

$$P_4=18 \text{ s}$$

$$P_1=20 \text{ s} ; P_5=8 \text{ s}$$

$$P_2=20 \text{ s}$$

$$P_3=18 \text{ s}$$

$$P_4=14 \text{ s}$$

b) Burst time kelima proses secara berturut-turut :  $(2, 4, 6, 8, 10)$  satuan waktu.

Turnaround time :

$$P_1=2 \text{ s} ; P_2=6 \text{ s}$$

$$P_3=10 \text{ s} ; P_4=14 \text{ s}$$

$$P_5=18 \text{ s}$$

Waiting time :

$$P_1=0 \text{ s} ; P_2=8 \text{ s}$$

$$P_3=2 \text{ s} ; P_4=5 \text{ s}$$

$$P_5=2 \text{ s}$$

- a) Kapan dimulai eksklusi top proses  
 $P_1=0 ; P_2=3 ; P_3=5 ; P_4=8 ; P_5=11$
- b) kapan selesai eksklusi  
 $t_1=8 ; P_2=5 ; P_3=8 ; P_4=16 ; P_5=12$

c) Total turnaround time

$$8+5+8+16+12=43$$

d) total min. TA

$$43/5=8.6$$

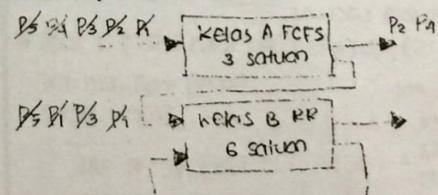
$$\begin{array}{l} \text{a)} \text{min. TA} \\ \text{b)} \text{total TA} \\ \text{c)} \text{Total turnaround time} \\ \text{d)} \text{min. TA} \end{array}$$

### 5.) Penjadwalan Proses Bertingkat

\* Soal 1 (Penjadwalan proses bertingkat (UTS 2010))

Sebuah sistem preemptive yang terdiri dari 2 kelas penjadwalan bertingkat : kelas A dan kelas B. kedua penjadwalan tersebut berfungsi secara bergantian dengan perbandingan 4:1. setiap CPU burst baru akan dieksekusi secara FCFS oleh penjadwalan kelas A. burst tidak rampung dalam 3 satuan waktu, akan dialihkan ke penjadwalan kelas B yang berbasis Round Robin dengan ketentuan 6 satuan waktu. Abukan waktu diukur.

$P_1(0:13), P_2(2:1), P_3(4:5), P_4(6:1), P_5(8:5) . P_x(Y:Z)$  burst proses X, miliki satuan Y selama Z satuan waktu.



0	1	2	3	4	5	6	7	8	9	10	11	12	
$P_1$	$A_{13}$	$A_{12}$	$A_{11}$	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$B_{10}$	$B_9$	$B_8$	$B_7$	$B_6$
$P_2$	$W_1$	$A_1$	—	—	—	—	—	—	—	—	—	—	—
$P_3$	$A_3$	$A_4$	$A_3$	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$	$W_{10}$
$P_4$	$W_1$	$A_1$	—	—	—	—	—	—	—	—	—	—	—

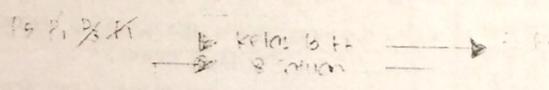
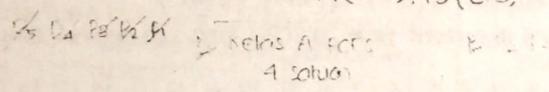
13	14	15	16	17	18	19	20	21	22	23	24	25
$P_1$	$B_5$	$W_7$	$W_8$	$W_9$	$W_{10}$	$B_2$	$B_1$	—	—	—	—	—
$P_2$	—	—	—	—	—	—	—	—	—	—	—	—
$P_3$	$W_7$	$W_8$	$W_9$	$W_{10}$	$B_2$	$B_1$	—	—	—	—	—	—
$P_4$	—	—	—	—	—	—	—	—	—	—	—	—
$P_5$	$W_6$	$A_5$	$A_4$	$A_3$	$W_7$	$W_8$	$W_9$	$W_{10}$	$W_{11}$	$W_{12}$	$B_2$	$B_1$

$$W(P_1)=10 ; W(P_2)=1 ; W(P_3)=1 ; W(P_4)=1 ; W(P_5)=12$$

### Soal 2 (Penjadwalan Proses Bertingkat (UTS 2009))

Sebuah sistem preemptive terdiri dari 2 kelas penjadwalan bertingkat: A dan B. Kedua penjadwalan tersebut berfungsi secara bergantian dengan perbandingan 3:1. Setiap CPU burst baru akan dieksekusi secara FCFS oleh penjadwalan kelas A. Burst tidak rampung dalam 4 satuan waktu akan dialihkan ke penjadwalan kelas B yang berbasis RR dengan quantum 2 satuan waktu.

P<sub>1</sub>(0:13), P<sub>2</sub>(2:1), P<sub>3</sub>(4:5), P<sub>4</sub>(6:1), P<sub>5</sub>(8:5)



	0	1	2	3	4	5	6	7	8	9	10	11	12
P <sub>1</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	P <sub>5</sub>
P <sub>2</sub>			W <sub>1</sub>	W <sub>2</sub>	A <sub>1</sub>	—	—	—	—	—	—	—	—
P <sub>3</sub>				W <sub>1</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	W <sub>6</sub>
P <sub>4</sub>					W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	W <sub>6</sub>	W <sub>7</sub>	W <sub>8</sub>	W <sub>9</sub>
P <sub>5</sub>						W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	W <sub>6</sub>	W <sub>7</sub>	W <sub>8</sub>

	13	14	15	16	17	18	19	20	21	22	23	24	25
P <sub>1</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	W <sub>6</sub>	W <sub>7</sub>	W <sub>8</sub>	W <sub>9</sub>	W <sub>10</sub>	W <sub>11</sub>	B <sub>1</sub>	—	—
P <sub>2</sub>	—	—	—	—	—	—	—	—	—	—	—	—	—
P <sub>3</sub>	W <sub>6</sub>	W <sub>7</sub>	W <sub>8</sub>	W <sub>9</sub>	W <sub>10</sub>	W <sub>11</sub>	W <sub>12</sub>	W <sub>13</sub>	W <sub>14</sub>	B <sub>1</sub>	—	—	—
P <sub>4</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	W <sub>6</sub>	A <sub>1</sub>	—	—	—	—	—	—	—	—
P <sub>5</sub>	W <sub>6</sub>	W <sub>7</sub>	W <sub>8</sub>	W <sub>9</sub>	W <sub>10</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	W <sub>11</sub>	W <sub>12</sub>	W <sub>13</sub>	W <sub>14</sub>

$W(P_1) = 11; W(P_2) = 2; W(P_3) = 14; W(P_4) = 11; W(P_5) = 12$

### Soal 3 (Penjadwalan Proses Bertingkat (2013))

Sebuah sistem preemptive terdiri dari tiga kelas penjadwalan bertingkat: kelas A, B dan C, dengan perbandingan 4:2:1. Setiap CPU Burst baru akan dieksekusi oleh penjadwalan kelas A yang berbasis FCFS dengan batasan 2 satuan waktu. Apabila burst tidak rampung dalam sekuat satuan, proses akan dialihkan ke penjadwalan kelas B yang berbasis FCFS dengan batasan 4 satuan waktu. Apabila burst belum rampung proses akan dialihkan ke penjadwalan C yang berbasis RR dengan quantum 6 satuan waktu.

P<sub>1</sub>(0:13), P<sub>2</sub>(2:2), P<sub>3</sub>(4:5), P<sub>4</sub>(6:1), P<sub>5</sub>(8:5).

	0	1	2	3	4	5	6	7	8	9	10	11	12
P <sub>1</sub>	A <sub>13</sub>	A <sub>12</sub>	W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	B <sub>11</sub>	B <sub>10</sub>	B <sub>9</sub>	B <sub>8</sub>	W <sub>6</sub>	W <sub>7</sub>
P <sub>2</sub>	A <sub>2</sub>	A <sub>1</sub>	—	—	—	—	—	—	—	—	—	—	—
P <sub>3</sub>			A <sub>5</sub>	A <sub>4</sub>	W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	B <sub>3</sub>	B <sub>2</sub>	—	—
P <sub>4</sub>					A <sub>1</sub>	—	—	—	—	—	—	—	—
P <sub>5</sub>						W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	W <sub>6</sub>	W <sub>7</sub>	W <sub>8</sub>

	13	14	15	16	17	18	19	20	21	22	23	24	25
P <sub>1</sub>	W <sub>8</sub>	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	W <sub>9</sub>	W <sub>10</sub>	C <sub>1</sub>	—	—	—
P <sub>2</sub>	—	—	—	—	—	—	—	—	—	—	—	—	—
P <sub>3</sub>	B <sub>1</sub>	—	—	—	—	—	—	—	—	—	—	—	—
P <sub>4</sub>	—	—	—	—	—	—	—	—	—	—	—	—	—
P <sub>5</sub>	W <sub>6</sub>	W <sub>7</sub>	W <sub>8</sub>	W <sub>9</sub>	W <sub>10</sub>	W <sub>11</sub>	W <sub>12</sub>	A <sub>5</sub>	A <sub>4</sub>	W <sub>13</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>

$W(P_1) = 15; W(P_2) = 0; W(P_3) = 5; W(P_4) = 0; W(P_5) = 13$

### STATUS PROSES

#### 1) Process State III (Kunyuk)

Terdapat 4 proses, P<sub>1</sub>(0:4,0), P<sub>2</sub>(10:4,9), P<sub>3</sub>(15:2,9), P<sub>4</sub>(20:3,3). (P<sub>i</sub>(A:B); A=starting time, B=CPU Time).

	1/10 wait = 60%	Menggunakan sisa	1	2	3	4
CPU Busy	40%	64%	72%	80%	88%	92%
CPU Busy for Access	40%	37%	33%	22%	17%	12%

	4	0
P <sub>1</sub>	4	—
P <sub>2</sub>	4,9	2,9 1,3 0
P <sub>3</sub>	2	1,6 1,3 0
P <sub>4</sub>	2,9	1,3 0

5 10 15 20 25 30

1) Waktu 0-10

$\text{hanya } P_1 \text{ yang jalannya} \rightarrow W_{CPU} = 4\% \times 10 = 40\% \times 10 = 4$

2) Waktu 10-15

$P_2 \text{ jalannya} \rightarrow W_{CPU} = 40\% \times 5 = 2$

$\text{Waktu sisa } P_2 = 4,9 - 2 = 2,9$

3) Waktu 15-20

$P_2 \& P_3 \text{ jalannya} \rightarrow W_{CPU} P_2 = 32\% \times 5 = 1,6$

$\text{Waktu sisa } P_2 = 2,9 - 1,6 = 1,3$

$\text{Waktu sisa } P_3 = 2,9 - 1,3 = 1,6$

4) Waktu 20-25

$P_2, P_3 \& P_4 \text{ jalannya} \rightarrow W_{CPU} = 26\% > 5 \rightarrow i.c.$

$\text{Waktu sisa } P_2 = 1,3 - 1,3 = 0$

$\text{Waktu sisa } P_3 = 1,3 - 1,3 = 0$

$\text{Waktu sisa } P_4 = 3,3 - 1,8 = 1,5$

5) Waktu 25-30

$P_4 \text{ sendiri} \rightarrow W_{CPU} = 40\% \times 5 = 2$

$\text{Waktu sisa} = 2 - 2 = 0$

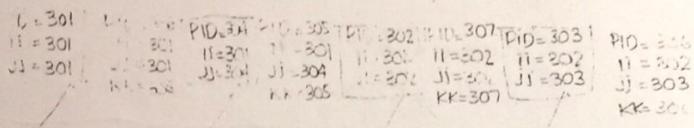
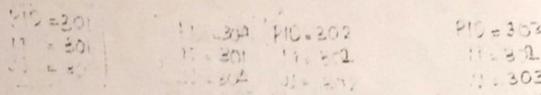
2) Process State (2010-1).

Diketahui empat proses, A(90:17,2), B(80:24,5), C(70:10,5), D(60:30); [W(X:Y)]  $\rightarrow W$ =nama proses, X=1/10 wait (%), Y=waktu (CPU) mulai saat bersamaan.

	Kombinasi Multiprogram (%)			
	A	A+B	A+B+C	A+B+C+D
Util CPU/proses A	10	9,3	8,1	7
—	B	—	19	16
—	C	—	—	21
—	D	—	—	32
				28



PID = 301



$i = 301 \dots j = 301 \dots k = 301$   
 $i = 301 \dots j = 302 \dots k = 302$   
 $i = 302 \dots j = 303 \dots k = 303$   
 $i = 301 \dots j = 304 \dots k = 304$   
 $i = 301 \dots j = 305 \dots k = 305$   
 $i = 302 \dots j = 306 \dots k = 306$   
 $i = 302 \dots j = 307 \dots k = 307$   
 $i = 301 \dots j = 308 \dots k = 308$   
 $i = 301 \dots j = 309 \dots k = 309$   
 $i = 301 \dots j = 310 \dots k = 310$   
 $i = 302 \dots j = 311 \dots k = 311$   
 $i = 302 \dots j = 312 \dots k = 312$

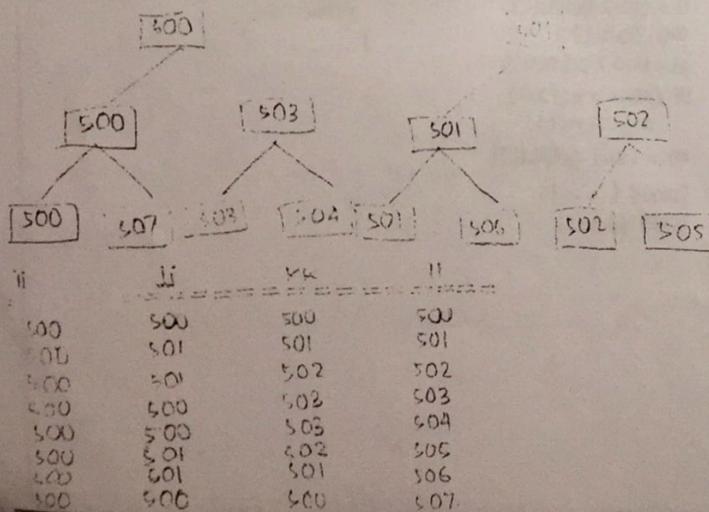
### \* Soal 2 (2011-1)

```

main() {
    int ii, jj, kk, ll;
    ii = jj = kk = ll = (int) getpid();
    printf(...);
    fork(2sec());
    jj = (int) getpid();
    fork(2sec());
    kk = (int) getpid();
    fork(2sec());
    ll = (int) getpid();
    printf(...);
    wait(NULL);
}
  
```

pid\_t fork - 2sec()  
 pid\_t tmps  
 sleep(1);  
 tmp = fork();  
 sleep(1);  
 return tmp;

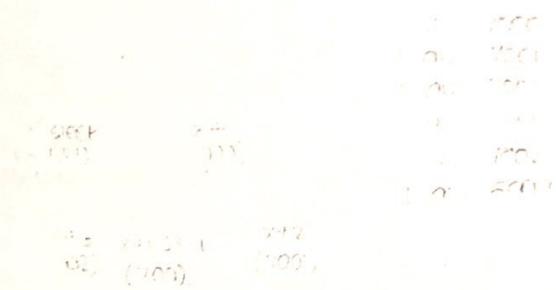
}



### \* Soal 3 (2011-2)

```

int main (int argc, char * argv[])
{
    int ii;
    for (ii=0; ii<2; ii++) {
        fork();
        waitpid(-1, NULL, 0);
        sleep(1);
        printf("I am %d\n", (int) getpid());
    }
}
  
```



### \* Soal 4 (2012-1)

```

#define STRING1 "PID [%5.5d] starts.\n"
#define STRING2 "PID [%5.5d] passes.\n"
#define STRING3 "PID [%5.5d] terminates.\n"
void main (void) {
    printf (STRING1, (int) getpid());
    fflush (stdout);
    for (int ii=0; ii<2; ii++) {
        pid_t pid1 = fork();
        if (pid1 == 0) {
            fork();
            wait (NULL);
            printf (STRING2, (int) getpid());
        }
        else {
            wait (NULL);
            printf (STRING3, (int) getpid());
        }
    }
    printf (STRING1, (int) getpid());
}
  
```

}

5001 X

5001 (21) X

5002 X

5001 (21) X

5005 (21) X

5002 (21) X

5003 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5006 (21) X

5001 (21) X 5008 (21) Y 5003 (21) X 5