Nama : Ikhsan Yusuf

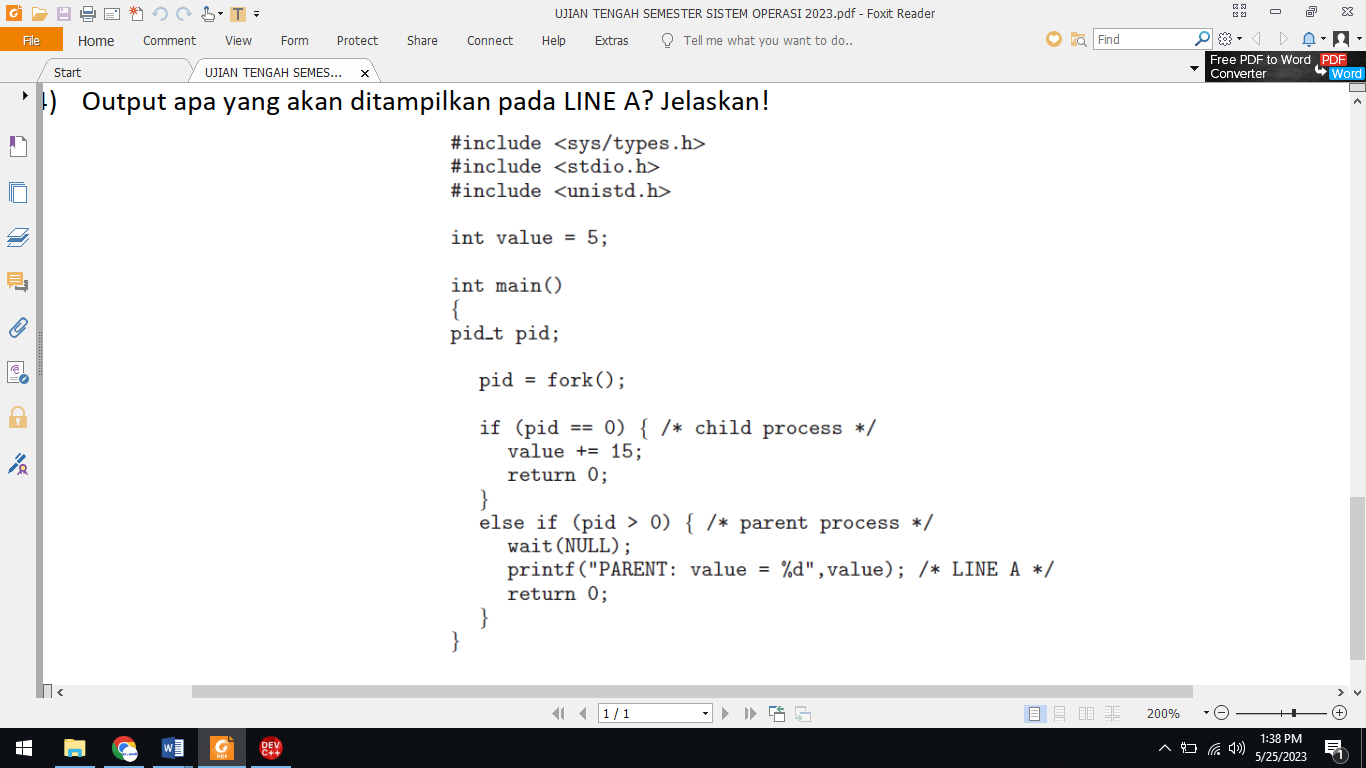
NIM : 2009076008

Mata Kuliah : Sistem Operasi

Ujian Tengah Semester

Soal

1. Jika diketahui 6 antrian proses (A, B, C, D, E, F) dengan waktu kedatangan secara bersamaan yaitu : 0. Lama eksekusi tiap-tiap antrian proses secara berurutan 1, 3, 7, 5, 5, 3. Hitunglah Trun Arround Time (TA) dengan menggunakan teknik penjadwalan proses:
   1. First In First Out (FIFO)
   2. Shortest Job First (SJF)
   3. Round Robin jika diketahui Quantum = 2
2. Dalam penjadwalan proses terdapat tiga macam penjadwalan, sebutkan, dan jelaskan disertai gambar!
3. Sumber daya apa yang digunakan saat thread dibuat? Bagaimana mereka berbeda dari yang digunakan ketika suatu proses dibuat?
4. Output apa yang akan ditampilkan pada LINE A? Jelaskan!



Jawaban:

1. Proses penjadwalan
   1. First In First Out (FIFO)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama | Waktu Tiba | Lama | Mulai | Selesai | Turn Arround |
| A | 0 | 1 | 0 | 1 | 1 |
| B | 0 | 3 | 1 | 4 | 4 |
| C | 0 | 7 | 4 | 11 | 11 |
| D | 0 | 5 | 11 | 16 | 16 |
| E | 0 | 5 | 16 | 21 | 21 |
| F | 0 | 3 | 21 | 24 | 24 |

∑TA = 1 + 4 + 11 + 16 + 21 + 24 = 77 (milidetik)

Rata-Rata TA = 77 : 6 = 12,83 (milidetik)

(Gannt Chart)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | | B | | C | | D | | E | | F | |
| 0 | 1 |  | 4 |  | 11 |  | 16 |  | 21 |  | 24 |

* 1. Shorten Job First (SJF)

(Proses lama eksekusi diurutkan dari yang terkecil hingga yang terbesar)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama | Waktu Tiba | Lama | Mulai | Selesai | Turn Arround |
| A | 0 | 1 | 0 | 1 | 1 |
| B | 0 | 3 | 1 | 4 | 4 |
| C | 0 | 3 | 4 | 7 | 7 |
| D | 0 | 5 | 7 | 12 | 12 |
| E | 0 | 5 | 12 | 17 | 17 |
| F | 0 | 7 | 17 | 24 | 24 |

∑TA = 1 + 4 + 7 + 12 + 17 + 24 = 65 (milidetik)

Rata-Rata TA = 65 : 6 = 10,83 (milidetik)

(Gannt Chart)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | | B | | C | | D | | E | | F | |
| 0 | 1 |  | 4 |  | 7 |  | 12 |  | 1 |  | 24 |

* 1. Round Robin

(Gannt Chart)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | | B | | C | | D | | E | | F | | B | | C | | D | | E | | F | | C | | D | | E | | C | |
| 0 | 1 |  | 3 |  | 5 |  | 7 |  | 9 |  | 11 |  | 12 |  | 14 |  | 16 |  | 18 |  | 19 |  | 21 |  | 22 |  | 23 |  | 24 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama | Waktu Tiba | Lama | Mulai | Selesai | Turn Arround |
| A | 0 | 1 | 0 | 1 | 1 |
| B | 0 | 3 | 1 | 12 | 12 |
| C | 0 | 7 | 3 | 24 | 24 |
| D | 0 | 5 | 5 | 22 | 22 |
| E | 0 | 5 | 7 | 23 | 23 |
| F | 0 | 3 | 9 | 19 | 19 |

∑TA = 1 + 12 + 24 + 22 + 23 + 19 = 101 (milidetik)

Rata-Rata TA = 101 : 6 = 16.83 (milidetik)

1. Terdapat tiga penjadwalan dalam sistem operasi yaitu
   1. Penjadwalan Jangka Pendek (*Short* *Term* *Scheduller*)

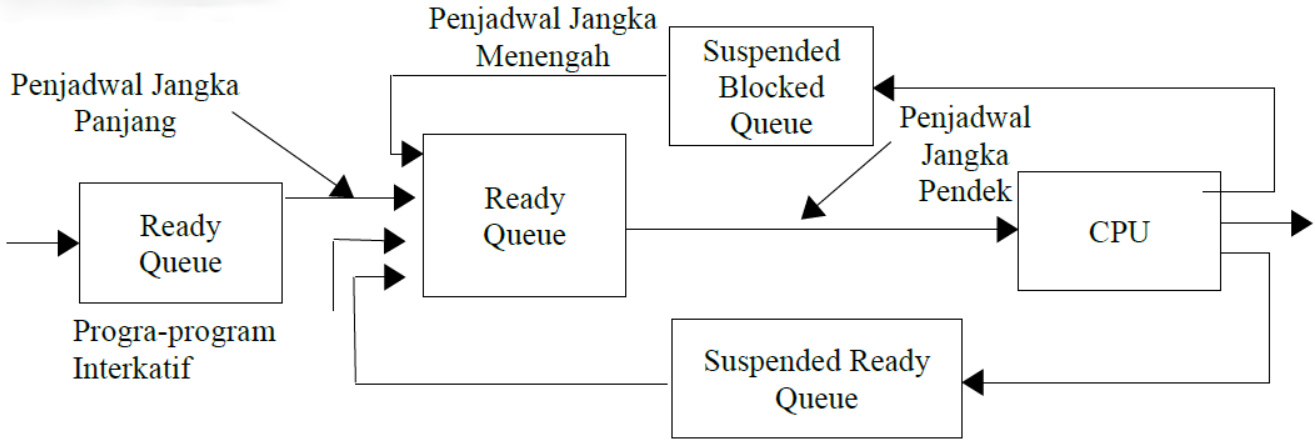
Memiliki tugas merencanakan pengalokasian pemroses antar proses yang akan dijalankan di memori utama.

* 1. Penjadwalan Jangka Menengah (*Medium Term* *Scheduller*)

Proses dipindahkan dari memori utama ke memori sekunder untuk memberi ruang kepada proses lain.

* 1. Penjadwalan Jangka Panjang (*Long Term* *Scheduller*)

Penjadwalan bekerja dengan antrean *batch* dan akan memilih *batch* berikutnya yang akan di eksekusi



1. Karena nilai *thread* lebih kecil dari proses, maka pada umumnya pembuatan *thread* akan menggunakan *resouce* yang sedikit dari pada saat pembuatan proses. Untuk membuat suatu proses, maka akan dibutuhkan alokasi PCB dan struktur data yang lebih besar. PCB sendiri terdiri atas peta memori, kumpulan daftar terbuka, dan juga variabel lingkungan.
2. *Output* akan berupa teks “PARENT : *Value* = 5

Segera setelah kita memanggil *fork*(), baik *child* dan *parent* berbagi data dan kode yang sama, dan ruang alamat *parent* disalin ke ruang alamat *child*. Ketika salah satu dari *child* atau *parent* mencoba untuk mengubah data (kode ditempatkan di bagian *read*-*only* dari *file* sehingga tidak dapat diubah), sebuah interupsi dibuat dan halaman ditulis untuk proses *child*, sebelum *parent* atau *child* dapat melakukan perubahan apa pun. Jadi pada akhirnya, *parent* dan *child* memiliki variabel mereka sendiri. Variabel yang dimiliki oleh *child* akan memiliki nilai 20 dan yang dimiliki oleh *parent* adalah 5