

Tugas 5

Sistem Operasi



Nama : Rheza Dewangga Rendragraha

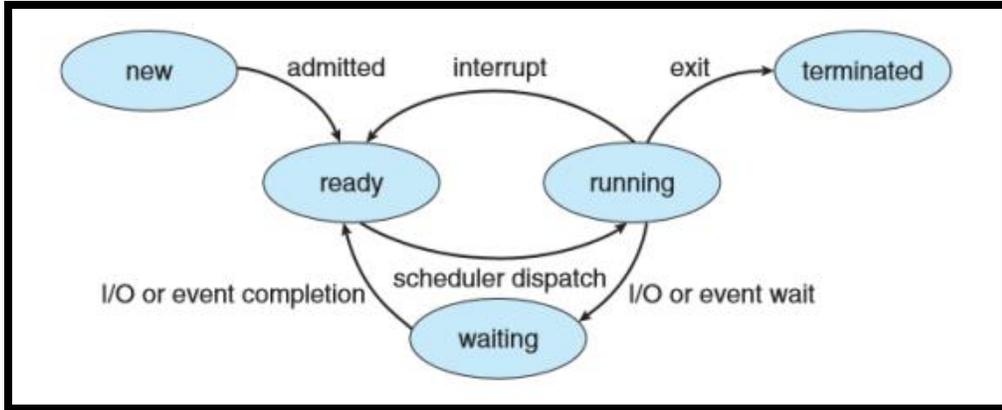
Kelas: 1 D4 Teknik Informatika B

NRP: 2110191044

Processes

Soal

1. Gambar ulanglah dan jelaskan mengenai diagram proses!



(GAMBAR 1)

2. Perbaikilah struktur diagram pada bab PCB!

process state
process number
program counter
registers
memory limits
list of open files
...

3. Eksekusi program berikut ini!

```
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

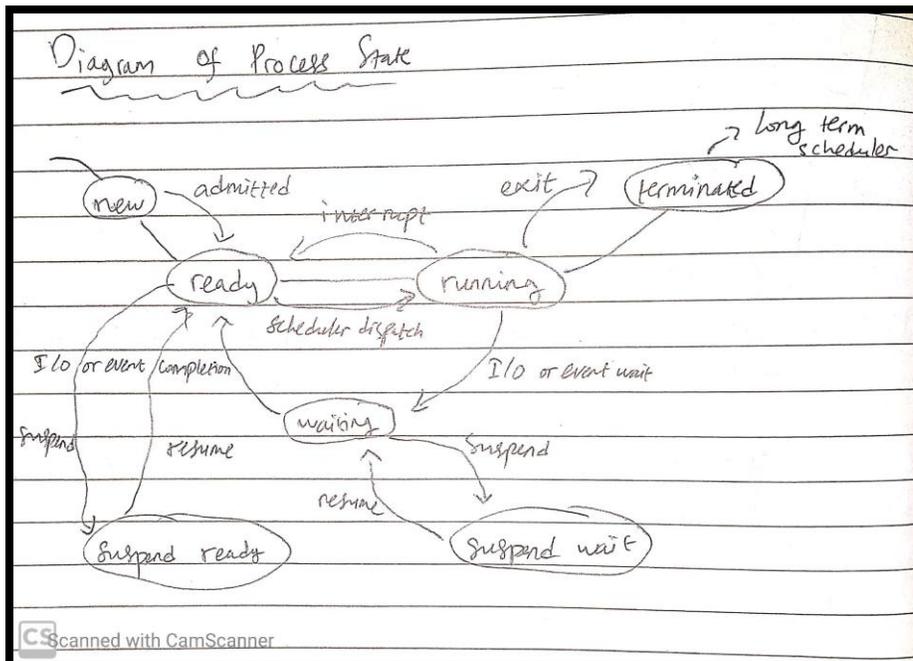
int main()
{
    pid_t pid;
    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occured */
        fprintf (stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        execlp("/bin/ls", "ls", NULL);
    }
    else { /* parent process */
        /* parent will wait for the child to complete */
        wait(NULL);
        printf("Child Complete");
    }

    return 0;
}
```

Jawab

1.



(GAMBAR 2)

Penjelasan

Pada diagram sebelumnya (Gambar 1) di soal, hanya terdapat 5 state of process, namun ternyata ada kondisi yang memungkinkan sehingga jumlah state of process bertambah menjadi 7, dengan 2 tambahan proses (Gambar 2), yaitu ketambahan:

- **Suspend wait**

State ini kemungkinan dilakukan ketika pada state waiting, CPU tidak mampu menampung semua proses, sehingga proses akan dibawa ke secondary memory. Jika proses pada state ini tidak membutuhkan I/O dan kondisi pada waiting memungkinkan, maka proses langsung dibawa ke main memory lagi ke state waiting.

- **Suspend ready**

State ini kemungkinan dilakukan ketika CPU saat main memory tidak mampu menampung proses pada state ready dalam artian pada state ready sudah mencapai kapasitas maksimum/degree of multiprogramming maximum. Ketika CPU sudah memungkinkan untuk menampung proses baru, maka proses yang ada pada suspend ready akan dibawa ke main memory lagi. Selain itu, suspend ready dimungkinkan dilakukan ketika proses pada suspend wait telah memenuhi kebutuhan seperti I/O. Tempat untuk menampung proses new state, ready state, waiting state, running state, dan terminated state yaitu di main memory. Untuk suspend ready state dan suspend wait state di secondary memory. Perlu diketahui juga bahwa diagram proses ini sangat berkaitan dengan Process Scheduler. Scheduler inilah yang bertanggung jawab dalam mentransfer sebuah proses dari state a ke state b. Ada 3 metode penjadwalan yang terjadi pada diagram proses tersebut, yaitu:

- **Long term scheduler atau Job Scheduler**

Scheduler ini bertanggung jawab dalam mengontrol degree of multiprogramming, yaitu berapa banyak proses yang dapat ditampung pada state ready. Dalam diagram proses, long term scheduler ini berada pada new state ke ready state. Job scheduler juga berperan dalam menciptakan keseimbangan antara I/O bound dengan CPU bound sehingga performancenya terjaga.

- **Short term scheduler atau CPU Scheduler**

Scheduler ini bertugas dalam penjadwalan proses dari ready state ke running state. Proses penjadwalan ini haruslah dieksekusi dengan cepat agar kinerja CPU berjalan dengan maksimal.

- **Medium/Middle term scheduler**

Scheduler ini diperlukan ketika suatu proses dijadwalkan dari running state ke ready state atau ke waiting state. Scheduler ini bertanggung jawab untuk memindahkan proses dari main memory ke secondary memory. Dalam diagram proses, medium term scheduler berperan ketika proses berada di suspend state (waiting maupun ready). Tujuan scheduler ini untuk

mempertahankan keseimbangan antara I/O bound dan CPU bound serta mengurangi jumlah degree of multiprogramming.

2.

Process State
Process Number
Process Privilege
Program Counter
Register
Scheduling Information
Memory Limits
Accounting Information
I/O Status Information
List of open files

Penjelasan

PCB (Process Control Block) adalah tempat penyimpanan semua informasi, semua metadata proses yang berjalan di sistem operasi mulai dari keadaan suatu proses hingga resources yang dibutuhkan. Berikut hierarki dari PCB:

- **Process state**
Berisi mengenai informasi status proses (ready, running, dll).
- **Process Privilege**
Berisi mengenai kepemilikan proses yang sedang berjalan.
- **Program counter**
Berisi informasi mengenai kondisi instruksi pada proses yang dijalankan itu sampai baris ke berapa.
- **CPU registers**
berkaitan dengan program counter, data yang sedang disimpan di CPU
- **CPU scheduling information**
Berisi mengenai informasi penjadwalan suatu proses termasuk prioritas sebuah proses.
- **Memory management/Memory Limits**
Termasuk memory limits berisi mengenai informasi maksimum memori yang dialokasikan untuk menjalankan sebuah proses.
- **Accounting information**
Berisi mengenai informasi penggunaan CPU, durasi pemakaian sistem.
- **I/O status information**
Berisi informasi mengenai penggunaan I/O device yang dialokasikan untuk menjalankan sebuah proses.

3. Sebelum diberi library #include <sys/wait.h>

```
user@user-VirtualBox:~$ gcc ch3.c
ch3.c: In function 'main':
ch3.c:21:3: warning: implicit declaration of function 'wait'; did you mean 'main'? [-Wimplicit-function-declaration]
    wait(NULL);
    ^
    main
user@user-VirtualBox:~$
```

Penjelasan

Program ini digunakan untuk menerapkan penggunaan fungsi fork() yang digunakan untuk membuat proses baru. Pada awalnya, program tersebut saya jalankan dengan terdapat pesan error yang menyatakan bahwa fungsi wait() tidak dikenali.

Lalu saya tambahkan library #include <sys/wait.h> dan program berjalan. Tipe data pid_t berfungsi untuk merepresentasikan PID. Program tersebut mengilustrasikan UNIX system call. Di dalam UNIX Shell akan membaca perintah dari terminal, kemudian perintah tersebut di fork() sehingga menghasilkan proses anak yang nantinya akan mengeksekusi perintah dari shell tersebut, sedangkan proses induk hanya menunggu dengan menggunakan system call wait() dan mengeksekusi perintah lain saat proses anak mengalami terminasi.