BINUS UNIVERSITY ONLINE

People Innovation Excellence

Tugas Kelompok ke-1

Week 4

Group 1

Giorgio Wilson Wong 2902692705

Cornelius Vito Satya Jalasena 2902708016

Farrel Nikoson 2902704895

Wahyu Tri Anggoro 2902692554

Caren Wong 2902693222

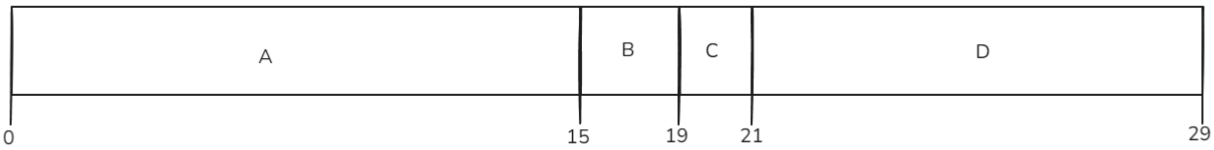
1. Misalkan dalam sebuah system, 4 proses sedang berjalan dengan waktu dating dan waktu proses sbb: (40%)

Process	CPU Burst Time	Arrival Time
A	15	0
B	4	2
C	2	7
D	8	10

Buatkan Gantt Chart dan hitunglah rata-rata waktu proses dan rata-rata waktu tunggu menggunakan algoritma penjadwalan:

a. **First Come First Serve**
Order : A > B > C > D

Process	**Start**	**Finish**	**Turnaround (Waktu Proses)**	**Waiting (Waktu Tunggu)**
A	0	0 + 15 = 15	15 - 0 = 15	15 - 15 = 0
B	15	15 + 4 = 19	19 - 2 = 17	17 - 4 = 13
C	19	19 + 2 = 21	21 - 7 = 14	14 - 2 = 12
D	21	21 + 8 = 29	29 - 10 = 19	19 - 8 = 11

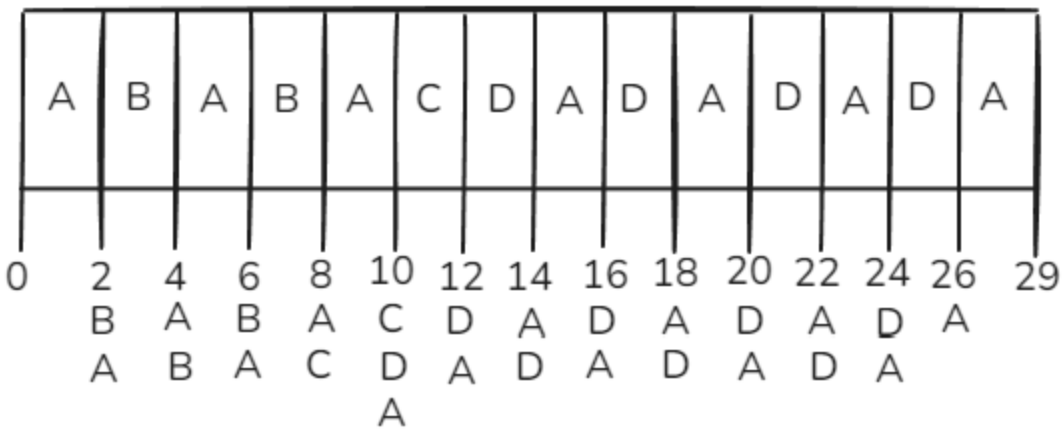


Rata-rata waktu proses
(15 + 17 + 14 + 19) / 4 = 16.25
Rata-rata waktu tunggu
(0 + 13 + 12 + 11) / 4 = 9

b. Shortest Job First (non-preemptive)
c. Shortest Job first (preemptive)
d. Round Robin dengan time slice=2

Order:

Time	Process	Burst Left	Ready in Queue
0-2	A	A = 13	B, A
2-4	B	B = 2	A, B
4-6	A	A = 11	B, A
6-8	B	B = 0	A, C
8-10	A	A = 9	C, D, A
10-12	C	C = 0	D, A
12-14	D	D = 6	A, D
14-16	A	A = 7	D, A
16-18	D	D = 4	A, D
18-20	A	A = 5	D, A
20-22	D	D = 2	A, D
20-24	A	A = 3	D, A
24-26	D	D = 0	A
26-29	A	A = 0	-



Process	Finish (Last Seen)	Turnaround (Waktu Proses)	Waiting (Waktu Tunggu)
A	29 (26-29)	29 - 0 = 29	29 - 15 = 14
B	8 (6-8)	8 - 2 = 6	6 - 4 = 2
C	12 (10-12)	12 - 7 = 5	5 - 2 = 3
D	26 (24-26)	26 - 10 = 16	16 - 8 = 8

Rata-rata waktu proses
(29 + 6 + 5 +16) / 4 = 14
Rata-rata waktu tunggu
(14 + 2 + 3 + 8) / 4 = 6.7

2. Jelaskan bagaimana proses state diimplementasikan pada system UNIX (15%)

Implementasi Process State pada Sistem UNIX
Pada sistem operasi UNIX, setiap proses memiliki kondisi tertentu yang disebut process state. Process state ini digunakan oleh sistem untuk mengetahui status dari suatu proses pada saat tertentu. Pengelolaan state proses dilakukan oleh kernel dengan menggunakan struktur data yang dikenal sebagai Process Control Block (PCB).

Beberapa process state utama pada sistem UNIX antara lain:
1. New
Proses berada pada kondisi ini saat baru saja dibuat, biasanya melalui system call seperti fork(). Kernel akan mengalokasikan PCB dan sumber daya awal yang dibutuhkan proses.
2. Ready
Proses sudah siap untuk dieksekusi tetapi masih menunggu giliran CPU. Proses ini ditempatkan di ready queue dan akan dipilih oleh scheduler.

3. Running
Proses sedang dieksekusi oleh CPU. Pada sistem single core, hanya satu proses yang dapat berada pada state ini dalam satu waktu.
4. Waiting / Blocked
Proses tidak dapat melanjutkan eksekusi karena menunggu suatu event, seperti operasi input/output, sinyal, atau resource tertentu.
5. Terminated
Proses telah selesai dieksekusi atau dihentikan. Kernel akan membersihkan resource yang digunakan oleh proses tersebut. Perpindahan antar process state dikendalikan oleh scheduler dan interrupt handler. Sebagai contoh, proses dapat berpindah dari running ke waiting saat melakukan operasi I/O, atau dari running ke ready ketika jatah waktu CPU habis

3. Jelaskan cara kerja system call fork() dan pthread_create(). Mengapa pembuatan thread lebih menguntungkan? Jelaskan. (15%)

Di sistem operasi UNIX, setiap proses pastinya berjalan dalam suatu process state. State ini dipakai oleh kernel untuk mengatur bagaimana dan kapan suatu proses dieksekusi oleh CPU.Secara konsep, process state ini diimplementasikan dan dikelola melalui struktur data internal kernel yang disebut Process Control Block (PCB).

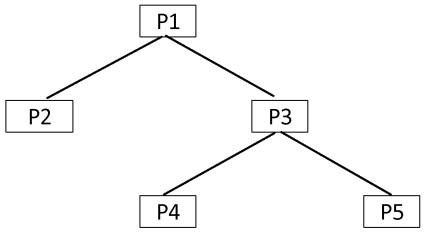
Beberapa process state utama pada UNIX antara lain New, Ready, Running, Waiting (Blocked),dan Terminated. Ketika sebuah proses baru dibuat (misalnya melalui fork()), proses tersebut masuk ke state New. Setelah semua resource dasar tersedia, proses akan masuk ke state Ready, artinya siap dijalankan tapi masih menunggu giliran CPU.

Saat scheduler memilih proses tersebut, statusnya berubah menjadi Running. Namun, proses tidak selalu bisa terus berjalan. Jika proses membutuhkan I/O (seperti membaca file atau menunggu input), maka proses akan berpindah ke state Waiting.Di state ini, proses dihentikan sementara sampai event yang ditunggu selesai. Setelah itu, proses akan kembali ke state Ready.

Apabila proses selesai dieksekusi atau dihentikan,maka proses masuk ke state Terminated dan resource yang digunakan akan dilepaskan oleh sistem. Perpindahan antar state ini sepenuhnya dikontrol oleh kernel UNIX menggunakan mekanisme context switching agar sistem tetap efisien dan stabil. Dengan adanya manajemen process state seperti ini,UNIX mampu menjalankan banyak proses secara bersamaan (multitasking) tanpa saling mengganggu, serta memastikan penggunaan CPU dan resource sistem tetap optimal.

Source GPT

4. Buatkan suatu program menggunakan fork() system call. Dalam program tsb. Buatkan proses-proses baru sehingga membentuk hirarki sbb: (30%)




```c
#include <stdio.h>
#include <unistd.h>

int main() {
    pid_t p2 = fork();
    if (p2 == 0) {
        printf("Hello from p2\n");
        return 0;
    }

    pid_t p3 = fork();
    if (p3 == 0) {
        printf("Hello from p3\n");

        pid_t p4 = fork();
        if (p4 == 0) {
            printf("Hello from p4\n");
            return 0;
        }

        pid_t p5 = fork();
        if (p5 == 0) {
            printf("Hello from p5\n");
            return 0;
        }

        return 0;
    }

    printf("Hello from p1\n");
    return 0;
}
```

Referensi

- BSD 4. fork(2) System Calls Manual. 4 June 1993. Diakses 14 Desember 2025.

- PPT04 Process Scheduling D6880 - EKO CAHYO NUGROHO, S.Kom., M.T.I. (Diakses 14/12/2025)