
Started on Thursday, 4 November 2021, 5:14 PM

State Finished

Completed on Thursday, 4 November 2021, 6:24 PM

Time taken 1 hour 10 mins

Marks 110.00/110.00

Grade **100.00** out of 100.00

Time limit	1 s
Memory limit	64 MB

Round robin adalah sebuah mekanisme yang digunakan dalam penjadwal (*scheduler*) proses atau jaringan. Setiap proses diberikan kesempatan untuk diproses dalam potongan waktu (*time slice*) yang sama, misalnya T_Q , dan jika proses tersebut masih membutuhkan waktu yang lebih lama untuk diproses, maka proses tersebut "diantrikan" kembali di urutan paling belakang.

Contoh:

Jika proses A membutuhkan waktu sebesar 14 satuan waktu untuk diselesaikan atau $T_A = 14$, dan jika $T_Q = 5$, maka proses A akan dieksekusi 3 kali, yaitu:

1. Eksekusi-1: selama 5 satuan waktu atau sebesar T_Q . Proses A kembali diantrikan dengan sisa waktu $T_A = T_A - T_Q = 14 - 5 = 9$.
2. Eksekusi-2: selama 5 satuan waktu atau sebesar T_Q . Selanjutnya, proses A kembali diantrikan dengan sisa waktu $T_A = T_A - T_Q = 9 - 5 = 4$.
3. Eksekusi-3: selama 4 satuan waktu atau T_A terkini (karena sudah $\leq T_Q$). Proses A selesai dieksekusi.

Dalam soal ini, mekanisme *round robin* diterapkan dalam suatu antrian yang dimodelkan dengan **list sirkuler** (file header ADT: [list_circular.h](#)). Setiap elemen list merepresentasikan waktu sisa pemrosesan suatu proses di komputer.

Buatlah sebuah program yang digunakan untuk mensimulasikan pemrosesan *round robin* dengan list sirkuler (file: **roundrobin.c**). Berikut adalah hal-hal yang dilakukan dalam program utama:

1. Menerima masukan sebuah integer > 0 yang merepresentasikan besarnya *time slice* untuk pemrosesan atau T_Q . Nilai T_Q harus divalidasi, yaitu dipastikan bahwa bernilai > 0 . Jika tidak, maka proses pemasukan T_Q harus diulang sampai didapatkan nilai yang benar (tidak perlu ada pesan kesalahan jika nilai masukan salah).
2. Menerima sejumlah operasi insert/delete ke sebuah list sirkuler yang merepresentasikan pemrosesan round robin. Kode untuk operasi direpresentasikan dengan character: 'A', 'D', 'F' dengan:

- Kode = 'A': operasi insert

Setelah kode operasi, diikuti dengan masukan waktu proses. Masukan waktu proses harus > 0 . Jika masukan waktu proses ≤ 0 , maka proses tersebut diabaikan (tidak akan dimasukkan ke dalam list).

Jika masukan waktu proses benar, tambahkan waktu proses sebagai elemen pertama list sirkuler.

- Kode = 'D': operasi delete

Jika list tidak kosong, maka dilakukan pemrosesan elemen terakhir dari list secara round robin. Misalnya elemen terakhir adalah last, maka:

- o jika $\text{Info}(\text{last}) > T_Q$, tuliskan waktu pemrosesan yang terjadi, yaitu sebesar T_Q .

Selanjutnya, insert kembali elemen last sebagai elemen pertama list dengan $\text{Info}(\text{last}) = \text{Info}(\text{last}) - T_Q$.

Proses insert ini tidak memerlukan penghapusan elemen last. Karena list sirkuler, yang dilakukan adalah cukup memindahkan penunjuk First(L) ke elemen last.

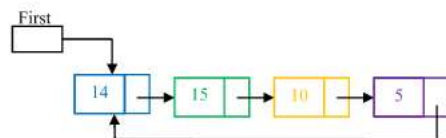
- o Jika $\text{Info}(\text{last}) \leq T_Q$, maka eksekusi proses tersebut telah selesai. Hapus elemen last (dan dealokasi) dan tuliskan waktu pemrosesan yang terjadi yaitu sebesar $\text{Info}(\text{last})$.

Jika list kosong, maka tuliskan pesan kesalahan "List kosong".

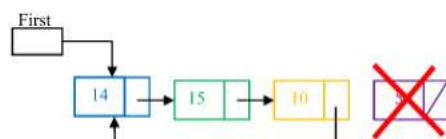
- Kode = 'F': selesai pemrosesan (keluar dari loop).
- Jika dimasukkan kode lain, maka tuliskan pesan "Kode salah".

Berikut adalah ilustrasi beberapa operasi delete dengan $T_Q = 5$:

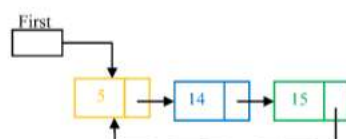
Time n: 4 elemen sudah di-insert



Time n+1: Operasi del (elemen ungu dihapus karena $\text{Info}(\text{elemen}) \leq T_Q$)



Time n+2: Operasi del (Penunjuk First pindah ke elemen oranye)



3. Setelah selesai, tuliskan waktu rata-rata proses yang masih tersisa di list jika list tidak kosong (presisi: dua angka di belakang koma). Jika list kosong, tuliskan "Proses selesai".

Untuk itu, buatlah fungsi dan prosedur berikut di program utama:

```
void deleteERR (List *l, ElType tq);
/* Jika l tidak kosong, "menghapus" elemen terakhir list l, misal last, sesuai aturan round robin, yaitu:
- jika INFO(last)>tq, maka last "diantrikan" kembali sebagai first elemen dengan INFO(last)=INFO(last)-tq
- jika INFO(last)<=tq, maka deleteLast(l,x) dan menampilkan x ke layar */
/* Jika l kosong, cetak "List kosong" */
/* I.S. tq terdefinisi sebagai nilai time slice pemrosesan.
L terdefinisi, mungkin kosong. */
/* F.S. Elemen terakhir l diproses sesuai aturan round-robin.
l mungkin menjadi kosong. */

float average (List l);
/* Menghasilkan nilai rata-rata elemen l. l tidak kosong. */
```

Contoh input/output (yang digaris bawah dan dicetak tebal adalah masukan pengguna):

Input/Output	Keterangan	Input/Output	Keterangan
<u>5</u>	TQ = 5	<u>10</u>	TQ = 10
<u>A</u>	Del-1: Elemen terakhir	<u>X</u>	Keadaan list
<u>5</u>	dihapus dengan waktu	Kode salah	setelah proses
<u>A</u>	pemrosesan = 5.	<u>A</u>	selesai: list
<u>10</u>	Del-2: Elemen terakhir	<u>10</u>	kosong
<u>A</u>	dengan waktu pemrosesan	<u>D</u>	
<u>15</u>	= 10, diproses = 5,	10	
<u>A</u>	diantrikan kembali = 5.	<u>D</u>	List kosong
<u>14</u>	Keadaan list setelah proses	<u>E</u>	Proses selesai
<u>D</u>	selesai: [5,14,15]		
5	Sehingga rata-rata = 11.33		
<u>D</u>			
5			
<u>E</u>			
11.33			
<u>-5</u>	TQ = 5 (validasi 2 kali)	<u>10</u>	TQ = 10
<u>0</u>	Del-1: Elemen terakhir	<u>X</u>	Keadaan list
<u>5</u>	dihapus dengan waktu	Kode salah	setelah proses
<u>A</u>	pemrosesan = 4.	<u>E</u>	selesai: list
<u>4</u>	...	Proses selesai	kosong
<u>A</u>	Del-4: Elemen terakhir		
<u>10</u>	dengan waktu pemrosesan		
<u>A</u>	= 14, diproses = 5,		
<u>15</u>	diantrikan kembali = 9.		
<u>D</u>	Del-5: Elemen terakhir		
4	dihapus dengan waktu		
<u>A</u>	pemrosesan = 5.		
<u>14</u>			
<u>D</u>	Keadaan list setelah proses		
5	selesai: [9,10]		
<u>D</u>	Sehingga rata-rata = 9.50		
5			
<u>D</u>			
5			
<u>D</u>			
5			
<u>E</u>			
9.50			

Yang harus di-submit: file [roundrobin.c](#)

C

 [roundrobin.c](#)

Score: 110

Blackbox

Score: 110

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.58 MB
2	10	Accepted	0.00 sec, 1.66 MB
3	10	Accepted	0.00 sec, 1.59 MB
4	10	Accepted	0.00 sec, 1.69 MB
5	10	Accepted	0.00 sec, 1.57 MB
6	10	Accepted	0.01 sec, 1.49 MB
7	10	Accepted	0.00 sec, 1.67 MB
8	10	Accepted	0.00 sec, 1.66 MB
9	10	Accepted	0.00 sec, 1.70 MB
10	10	Accepted	0.00 sec, 1.66 MB
11	10	Accepted	0.00 sec, 1.60 MB

[◀ Pra Praktikum 10](#)

Jump to...

[stack_linked.h ▶](#)