

Questão 01

```
$ _ linux@linux-PC:~/Documents/sistemas_operacionais/trabalho
→ trabalho git:(main) ls
q01.c q02.c q03a.c q03b.c q05 q05.c
→ trabalho git:(main) gcc q01.c -o q01 -lm -lpthread
→ trabalho git:(main) x ls
q01 q01.c q02.c q03a.c q03b.c q05 q05.c
→ trabalho git:(main) x ./q01
Digite um numero inteiro: 3
Total de primos = 2
→ trabalho git:(main) x ./q01
Digite um numero inteiro: 7
Total de primos = 4
→ trabalho git:(main) x ./q01
Digite um numero inteiro: 17
Total de primos = 7
→ trabalho git:(main) x ./q01
Digite um numero inteiro: 100
Total de primos = 25
→ trabalho git:(main) x
```

Questão 02

```
$ _ linux@linux-PC:~/Documents/sistemas_operacionais/trabalho
→ trabalho git:(main) ls
q01.c q02.c q03a.c q03b.c q05 q05.c
→ trabalho git:(main) gcc q02.c -o q02 -lm -lpthread
→ trabalho git:(main) x ./q02
Digite um valor para x: 2
0.909297
→ trabalho git:(main) x ./q02
Digite um valor para x: 0
0.000000
→ trabalho git:(main) x ./q02
Digite um valor para x: 1
0.841471
→ trabalho git:(main) x ./q02
Digite um valor para x: 4
-0.756800
→ trabalho git:(main) x ./q02
Digite um valor para x: 3
0.141120
→ trabalho git:(main) x ./q02
Digite um valor para x: -2
-0.909297
→ trabalho git:(main) x |
```

Questão 03a

```
$ _ linux@linux-PC:~/Documents/sistemas_operacionais/trabalho
→ trabalho git:(main) x ls
q01.c q02.c q03a.c q03b.c q05 q05.c
→ trabalho git:(main) x cc q03a.c
→ trabalho git:(main) x ./a.out
Pai → Filho → Filho1 → FIM
→ trabalho git:(main) x sS
```

Questão 03b

```
$ _ linux@linux-PC:~/Documents/sistemas_operacionais/trabalho
→ trabalho git:(main) x ls
* q01.c q02.c q03a.c q03b.c q05.c
i → trabalho git:(main) x cc q03b.c
i → trabalho git:(main) x ./a.out 1 2 3 4 5 6
i Eu sou o PAI e espero o FILHO 7246
i Sou FILHO 7246
i 1 → 2 → 3 → FIM FILHO
i Terminei, agora vou embora...
i O filho acabou
i Continua PAI
i 4 → 5 → 6 → FIM
i O Pai acabou
→ trabalho git:(main) x ./a.out 1 2 3 4 5 6 7 8 9
Eu sou o PAI e espero o FILHO 7284
Sou FILHO 7284
1 → 2 → 3 → 4 → 5 → FIM FILHO
Terminei, agora vou embora...
O filho acabou
Continua PAI
6 → 7 → 8 → 9 → FIM
O Pai acabou
→ trabalho git:(main) x |
```

Questão 05

Execução em uma VM linux com 2G RAM

```

$ _ linux@linux-PC:~/Documents/sistemas_operacionais/trabalho
→ trabalho git:(main) x ls
q01.c q02.c q03a.c q03b.c q05.c
→ trabalho git:(main) x gcc q05.c -o q05 -lm -lpthread
→ trabalho git:(main) x execução em uma VM linux com 2G RAM
→ trabalho git:(main) x ./q05
Produtor 0 produziu 83
Consumidor 4 consumiu 83
Consumidor 4 consumiu 86
Produtor 4 produziu 77
Consumidor 4 consumiu 77
Consumidor 4 consumiu 15
Consumidor 4 consumiu 15
Produtor 4 produziu 86
Produtor 4 produziu 93
Consumidor 4 consumiu 93
Consumidor 4 consumiu 35
Produtor 4 produziu 35
Produtor 4 produziu 86
Consumidor 4 consumiu 86
Consumidor 4 consumiu 92
Produtor 4 produziu 92
Produtor 4 produziu 49
Consumidor 4 consumiu 49
Consumidor 4 consumiu 21
Consumidor 4 consumiu 92
Produtor 4 produziu 21
Produtor 4 produziu 62
Consumidor 4 consumiu 77
Consumidor 4 consumiu 27
Produtor 4 produziu 27
Produtor 4 produziu 90

```

```

$ _ linux@linux-PC:~/Documents/sistemas_operacionais/trabalho
Produtor 4 produziu 62
Consumidor 4 consumiu 77
Consumidor 4 consumiu 27
Produtor 4 produziu 27
Produtor 4 produziu 90
Consumidor 4 consumiu 90
Produtor 4 produziu 15
Produtor 4 produziu 59
Consumidor 4 consumiu 62
Consumidor 4 consumiu 63
Produtor 4 produziu 63
Produtor 4 produziu 26
Consumidor 4 consumiu 26
Consumidor 4 consumiu 40
Produtor 4 produziu 26
Consumidor 4 consumiu 63
Consumidor 4 consumiu 72
Produtor 4 produziu 72
Produtor 4 produziu 36
Produtor 4 produziu 40
Consumidor 4 consumiu 72
Consumidor 4 consumiu 11
Produtor 4 produziu 11
Produtor 4 produziu 68
Consumidor 4 consumiu 68
Consumidor 4 consumiu 67
Produtor 4 produziu 67
Produtor 4 produziu 29

FIM
→ trabalho git:(main) x |

```

```

$ _ linux@linux-PC:~/Documents/sistemas_operacionais/trabalho
→ trabalho git:(main) x ./q05
Produtor 0 produziu 83
Consumidor 4 consumiu 83
Consumidor 4 consumiu 86
Produtor 4 produziu 77
Consumidor 4 consumiu 77
Consumidor 4 consumiu 15
Produtor 4 produziu 93
Consumidor 4 consumiu 93
Produtor 4 produziu 35
Consumidor 4 consumiu 77
Consumidor 4 consumiu 86
Consumidor 4 consumiu 77
Produtor 4 produziu 92
Consumidor 4 consumiu 35
Produtor 4 produziu 86
Produtor 4 produziu 49
Produtor 4 produziu 15
Consumidor 4 consumiu 49
Consumidor 4 consumiu 21
Consumidor 4 consumiu 35
Produtor 4 produziu 21
Produtor 4 produziu 62
Consumidor 4 consumiu 27
Produtor 4 produziu 27
Produtor 4 produziu 90
Produtor 4 produziu 86

FIM
→ trabalho git:(main) x ss

```

```

$ _ linux@linux-PC:~/Documents/sistemas_operacionais/trabalho
FIM
→ trabalho git:(main) x ./q05
Produtor 3 produziu 83
Consumidor 4 consumiu 83
Consumidor 4 consumiu 86
Produtor 4 produziu 86
Produtor 4 produziu 77
Consumidor 4 consumiu 77
Consumidor 4 consumiu 15
Produtor 4 produziu 15
Produtor 4 produziu 93
Consumidor 4 consumiu 93
Consumidor 4 consumiu 35
Consumidor 4 consumiu 35
Produtor 4 produziu 35
Consumidor 4 consumiu 86
Produtor 4 produziu 92
Produtor 4 produziu 49
Consumidor 4 consumiu 86
Consumidor 4 consumiu 92
Produtor 4 produziu 21
Consumidor 4 consumiu 49
Consumidor 4 consumiu 86
Produtor 4 produziu 62
Produtor 4 produziu 86
Consumidor 4 consumiu 62
Consumidor 4 consumiu 21

FIM
→ trabalho git:(main) x

```

Execução em uma máquina windows 8G RAM

```

Produtor 3 produziu 41
Consumidor 1 consumiu 41
Produtor 2 produziu 41
Consumidor 2 consumiu 41
Produtor 4 produziu 67
Consumidor 4 consumiu 67
Produtor 4 produziu 34
Consumidor 4 consumiu 67
Consumidor 4 consumiu 34
Consumidor 4 consumiu 34
Consumidor 4 consumiu 34
for(int i=0; i<QTD; i++){
    pthread_join(pro[i], NULL);
    pthread_join(con[i], NULL);
}
Process returned 0 (0x0)   execution time : 0.019 s
Press any key to continue.
printf("\nFIM\n"); /*Para saber o FIM do main
return 0;

```

```

C:\Users\Student\Desktop\q05\bin\Debug\q05.exe
Produtor 4 produziu 41
Consumidor 4 consumiu 41
Consumidor 4 consumiu 41
Consumidor 4 consumiu 41
Consumidor 4 consumiu 67
Produtor 4 produziu 67
Consumidor 4 consumiu 67
Produtor 4 produziu 34
Produtor 4 produziu 0
Produtor 4 produziu 69
Consumidor 4 consumiu 69
Produtor 4 produziu 24
Consumidor 4 consumiu 24
Produtor 4 produziu 78
Consumidor 4 consumiu 78
for(int i=0; i<QTD; i++){
    pthread_join(pro[i], NULL);
    pthread_join(con[i], NULL);
}
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.
printf("\nFIM\n"); /*Para saber o FIM do main
return 0;

```

Questão 06

```

C:\Users\Student\Desktop\q06\bin\Debug\q06.exe
Quantos processos estao na fila: 3
Digite um identificador para o 1º processo(char): a
Digite a ordem de chegada do processo(int): 2
Digite o tempo de execucao do processo(int): 3
Duracao do processo(int): 3
Digite um identificador para o 2º processo(char): b
Digite a ordem de chegada do processo(int): 3
Digite o tempo de execucao do processo(int): 5
Duracao, ordem);
Digite um identificador para o 3º processo(char): c
Digite a ordem de chegada do processo(int): 1
Digite o tempo de execucao do processo(int): 1

[1] - SFJ
[2] - Round Robin
Qual algoritmo de escalonamento deseja: 2
quantum: 2
id: c executando...(1s) scanf("%d", &op);
id: a executando...(3s)
id: b executando...(5s)
id: a executando...(1s)
id: b executando...(3s)
id: b executando...(1s)
printf("\nQuantum");

```



```

C:\Users\Student\Desktop\q06\bin\Debug\q06.exe
Quantos processos estão na fila: 4
Digite um identificador para o 1º processo(char): a
Digite a ordem de chegada do processo(int): 2
Digite o tempo de execucao do processo(int): 3
Digite um identificador para o 2º processo(char): b
Digite a ordem de chegada do processo(int): 1
Digite o tempo de execucao do processo(int): 5
Digite um identificador para o 3º processo(char): c
Digite a ordem de chegada do processo(int): 4
Digite o tempo de execucao do processo(int): 1
Digite um identificador para o 4º processo(char): d
Digite a ordem de chegada do processo(int): 3
Digite o tempo de execucao do processo(int): 1
/*Menu para utilizacao*/
[1] - SFJ
[2] - Round Robin
Qual algoritmo de escalonamento deseja: 1
Ordem dos processos por prioridade: escalonamento deseja: "); scanf("%d",&op);
b -> a -> d -> c -> FIM
if(op == 1){
INICIO SJF
Processo c executando...(1s)
Processo d executando...(1s)
Processo a executando...(3s)
Processo b executando...(5s)
FIM SJF
round_robin(proc, quantum);
}else{
Process returned 0 (0x0)   execution time: 69.628 s
Press any key to continue.

```