

Tarefa 07

Alunas: Maria Luiza e Thayris Gabriela

Código completo: <https://github.com/thayrisgabriela/CDs>

7.0

```
o processo 0 vai testar no tempo 30.0
o processo 1 vai testar no tempo 30.0
o processo 1 falhou no tempo 31.0
o processo 0 vai testar no tempo 60.0
o processo 1 recuperou no tempo 61.0
o processo 1 vai testar no tempo 62.0
o processo 0 vai testar no tempo 90.0
o processo 1 vai testar no tempo 92.0
o processo 0 vai testar no tempo 120.0
o processo 1 vai testar no tempo 122.0
o processo 0 vai testar no tempo 150.0
```

7.1 e 7.2

```
while (time() < 150.0) {
    cause(&event, &token);
    switch (event) {
        case test: {
            int current_st = status(processo[token].id);
            if (current_st != 0) // erro no processo atual
                break;
            int prox_token = (token + 1) % N;
            printf("\nTempo: %5.1f\n", time());
            for (int i = 1; i < N; i++) {
                printf(" Processo %d testando o %d\n", token, prox_token);
                int st = status(processo[prox_token].id);
                if (st != 0) {
                    printf(" Erro no processo %d :(\n", prox_token);
                    // schedule(fault, 1.0, prox_token);
                    prox_token = (prox_token + 1) % N;
                } else {
                    // printf("SUCESSO: processo %d irá testar no tempo
                    // %5.1f\n", prox_token, time());
                    printf(" SUCESSO!\n");
                    schedule(test, 30.0, prox_token);
                }
            }
            break;
        }
    }
}
```

```
Tempo: 30.0
  Processo 0 testando o 1
    SUCESSO!

Tempo: 30.0
  Processo 1 testando o 2
    SUCESSO!

Tempo: 30.0
  Processo 2 testando o 3
    SUCESSO!

Tempo: 30.0
  Processo 3 testando o 4
    SUCESSO!

Tempo: 30.0
  Processo 4 testando o 5
    SUCESSO!

Tempo: 30.0
  Processo 5 testando o 0
    SUCESSO!

--Processo 1 falhou no tempo 31.0

Tempo: 60.0
  Processo 2 testando o 3
    SUCESSO!

Tempo: 60.0
  Processo 3 testando o 4
    SUCESSO!
```

Tempo: 60.0
Processo 4 testando o 5
SUCESSO!

Tempo: 60.0
Processo 5 testando o 0
SUCESSO!

Tempo: 60.0
Processo 0 testando o 1
Erro no processo 1 :(
Processo 0 testando o 2
SUCESSO!

Tempo: 90.0
Processo 3 testando o 4
SUCESSO!

Tempo: 90.0
Processo 4 testando o 5
SUCESSO!

Tempo: 90.0
Processo 5 testando o 0
SUCESSO!

Tempo: 90.0
Processo 0 testando o 1
Erro no processo 1 :(
Processo 0 testando o 2
SUCESSO!

```
Processo 0 testando o 1
  Erro no processo 1 :(
Processo 0 testando o 2
  SUCESSO!

Tempo: 90.0
  Processo 2 testando o 3
    SUCESSO!

Tempo: 120.0
  Processo 4 testando o 5
    SUCESSO!

Tempo: 120.0
  Processo 5 testando o 0
    SUCESSO!

Tempo: 120.0
  Processo 0 testando o 1
    Erro no processo 1 :(
  Processo 0 testando o 2
    SUCESSO!

Tempo: 120.0
  Processo 2 testando o 3
    SUCESSO!

Tempo: 120.0
  Processo 3 testando o 4
    SUCESSO!
```

7.3

```

while (time() < 150.0) {
    cause(&event, &token);
    switch (event) {
        case test: {
            int current_st = status(processo[token].id);
            if (current_st != 0) // erro no processo atual
                break;
            int prox_token = (token + 1) % N;
            printf("\nTempo: %5.1f\n", time());
            for (int i = 1; i < N; i++) {
                printf("  Processo %d testando o %d\n", token, prox_token);
                int st = status(processo[prox_token].id);
                STATE[token][prox_token] = st;
                if (st != 0) {
                    printf("    Erro no processo %d :(\n", prox_token);
                    // schedule(fault, 1.0, prox_token);
                    prox_token = (prox_token + 1) % N;
                } else {
                    // printf("SUCESSO: processo %d irá testar no tempo\n", prox_token, time());
                    printf("    SUCESSO!\n");
                    schedule(test, 30.0, prox_token);
                }
            }
            break;
        }
    }
}

```

```

Tempo: 120.0
  Processo 3 testando o 4
    SUCESSO!

Tempo: 150.0
  Processo 5 testando o 0
    SUCESSO!

Processo 0: 0 1 0 -1 -1 -1
Processo 1: -1 0 0 -1 -1 -1
Processo 2: -1 -1 0 0 -1 -1
Processo 3: -1 -1 -1 0 0 -1
Processo 4: -1 -1 -1 -1 0 0
Processo 5: 0 -1 -1 -1 -1 0

```

```

while (time() < 150.0) {
    cause(&event, &token);
    switch (event) {
        case test: {
            int current_st = status(processo[token].id);
            if (current_st != 0) // erro no processo atual
                break;
            int prox_token = (token + 1) % N;
            printf("\nTempo: %5.1f\n", time());
            for (int i = 1; i < N; i++) {
                printf("  Processo %d testando o %d\n", token, prox_token);
                int st = status(processo[prox_token].id);
                STATE[token][prox_token] = st;
                if (st != 0) {
                    printf("    Erro no processo %d :(\n", prox_token);
                    // schedule(fault, 1.0, prox_token);
                    prox_token = (prox_token + 1) % N;
                } else {
                    // printf("SUCESSO: processo %d irá testar no tempo\n", prox_token, time());
                    printf("    SUCESSO!\n");
                    schedule(test, 30.0, prox_token);
                }
            }
            for (int j = 0; j < N; j++) {
                if (STATE[prox_token][j] != -1) {
                    STATE[token][j] = STATE[prox_token][j];
                }
            }
            break;
        }
    }
}

```

```

SUCESSO!

Tempo: 120.0
  Processo 3 testando o 4
    SUCESSO!

Tempo: 150.0
  Processo 5 testando o 0
    SUCESSO!

Processo 0: 0 0 0 0 0 0
Processo 1: -1 0 0 -1 -1 -1
Processo 2: 0 0 0 0 0 0
Processo 3: 0 1 0 0 0 0
Processo 4: 0 1 0 0 0 0
Processo 5: 0 0 0 0 0 0

```