

Sistemas Operacionais

4º Semestre

Prof. Jamilson Bispo dos Santos

Nome: Gustavo Cunha Ciola

TIA: 32247141

Nome: Ricardo Veiga Coelho

TIA: 42241588

Nome: Vinicius Moreira Tamburu

TIA: 32227515

Projeto 02 - Escada Rolante Dupla

Códigos fonte:

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

```
int main() {  
    int N;  
    int t, d;  
    int totalTime = 0;
```

```
    printf("Digite o número de pessoas que usarão a escada rolante: ");  
    scanf("%d", &N);
```

```
    int dValues[N];  
    int tValues[N];
```

```
    printf("Digite o momento de chegada e a direção (0 para esquerda, 1 para  
    direita) para cada pessoa:\n");
```

```
    // Leitura dos valores de d e cálculo inicial do totalTime  
    for (int i = 0; i < N; i++) {  
        printf("Pessoa %d: ", i + 1);  
        scanf("%d %d", &t, &d);
```

```
        dValues[i] = d; // Armazena os valores de d  
        tValues[i] = t; // Armazena os valores de t
```

```
}
```

```
for (int i = 0; i < N; i++) {  
    if(dValues[i + 1] == dValues[i]){  
        totalTime += tValues[i + 1] - tValues[i];  
    }else{  
        int soma = tValues[i] - totalTime;  
        totalTime += soma + 10;  
    }  
}
```

```
}
```

```
printf("O momento em que a última pessoa saiu da escada: %d\n", totalTime);
```

```
return 0;
```

```
}
```

```

#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

typedef struct {
    int arrivalTime;
    int direction;
} Person;

sem_t mutex; // Semáforo para controlar o acesso à escada
int currentTime = 0; // Variável global para acompanhar o tempo atual
int lastPersonExitTime = 0; // Variável para armazenar o momento em que a última
pessoa saiu da escada

void *escalator(void *arg) {
    Person *person = (Person *)arg;

    // Espera até o momento de chegada da pessoa
    sleep(person->arrivalTime - currentTime);

    sem_wait(&mutex); // Bloqueia o acesso à escada

    // Simula a viagem na escada (leva 10 segundos)
    sleep(10);
    currentTime = person->arrivalTime + 10; // Atualiza o tempo atual após a pessoa
sair

    sem_post(&mutex); // Libera o acesso à escada
    lastPersonExitTime = currentTime; // Atualiza o momento em que a última pessoa
saiu da escada

    pthread_exit(NULL);
}

int main() {
    int numPeople;
    printf("Digite o número de pessoas que usarão a escada rolante: ");
    scanf("%d", &numPeople);

    Person people[numPeople];

    printf("Digite o momento de chegada e a direção (0 para esquerda, 1 para direita)
para cada pessoa:\n");

```

```
for (int i = 0; i < numPeople; i++) {
    printf("Pessoa %d: ", i + 1);
    scanf("%d %d", &people[i].arrivalTime, &people[i].direction);
}

sem_init(&mutex, 0, 1); // Inicializa o semáforo

pthread_t threads[numPeople];
for (int i = 0; i < numPeople; i++) {
    pthread_create(&threads[i], NULL, escalator, (void *)&people[i]);
}

for (int i = 0; i < numPeople; i++) {
    pthread_join(threads[i], NULL);
}

sem_destroy(&mutex); // Destrói o semáforo após o uso

printf("O momento em que a última pessoa saiu da escada é: %d\n",
lastPersonExitTime);

return 0;
}
```

Testes:

```
/media/ciola87/Extreme SSD/Mackenzie/4_sememstr... - □ ×
Digite o número de pessoas que usarão a escada rolante: 5
Digite o momento de chegada e a direção (0 para esquerda, 1 para direita) para cada pessoa:
Pessoa 1: 5 0
Pessoa 2: 8 0
Pessoa 3: 13 0
Pessoa 4: 25 1
Pessoa 5: 27 1
O momento em que a última pessoa saiu da escada é: 37
Process returned 0 (0x0)   execution time : 17,853 s
Press ENTER to continue.
□
```

```
/media/ciola87/Extreme SSD/Mackenzie/4_sememstr... - □ ×
Digite o número de pessoas que usarão a escada rolante: 6
Digite o momento de chegada e a direção (0 para esquerda, 1 para direita) para cada pessoa:
Pessoa 1: 5 0
Pessoa 2: 8 0
Pessoa 3: 13 0
Pessoa 4: 25 1
Pessoa 5: 30 1
Pessoa 6: 42 0
O momento em que a última pessoa saiu da escada é: 52
Process returned 0 (0x0)   execution time : 34,672 s
Press ENTER to continue.
□
```

```
/media/ciola87/Extreme SSD/Mackenzie/4_sememstr... - □ ×
Digite o número de pessoas que usarão a escada rolante: 3
Digite o momento de chegada e a direção (0 para esquerda, 1 para direita) para cada pessoa:
Pessoa 1: 5 0
Pessoa 2: 8 0
Pessoa 3: 13 0
O momento em que a última pessoa saiu da escada é: 23
Process returned 0 (0x0)   execution time : 11,414 s
Press ENTER to continue.
□
```

```
Digite o número de pessoas que usarão a escada rolante: 5
Digite o momento de chegada e a direção (0 para esquerda, 1 para direita) para cada
pessoa:
Pessoa 1: 5 0
Pessoa 2: 8 0
Pessoa 3: 13 0
Pessoa 4: 25 1
Pessoa 5: 27 1
O momento em que a última pessoa saiu da escada é: 37
```

```
Digite o número de pessoas que usarão a escada rolante: 6
Digite o momento de chegada e a direção (0 para esquerda, 1 para direita) para cada
pessoa:
Pessoa 1: 5 0
Pessoa 2: 8 0
Pessoa 3: 13 0
Pessoa 4: 25 1
Pessoa 5: 30 1
Pessoa 6: 42 0
O momento em que a última pessoa saiu da escada é: 52
```

```
Digite o número de pessoas que usarão a escada rolante: 3
Digite o momento de chegada e a direção (0 para esquerda, 1 para direita) para cada
pessoa:
Pessoa 1: 5 0
Pessoa 2: 8 0
Pessoa 3: 13 0
O momento em que a última pessoa saiu da escada é: 23
```

A primeira versão é uma implementação sequencial simples, enquanto a segunda versão utiliza threads e semáforos para simular concorrência entre as pessoas que utilizam a escada rolante.