

### **Projeto de Programação 01**

O trabalho pode ser feito em duplas. Este trabalho exercita as primitivas básicas da Linguagem C e inicia a discussão e solução de problemas complexos.

Você foi contratado pela Acima e Avante Elevadores para projetar a próxima geração de elevadores da empresa. De forma a reduzir os custos de desenvolvimento, você optou por implementar um simulador orientado a eventos do controlador de elevador. O ambiente de funcionamento de um elevador consiste do número de andares do prédio e da sua capacidade. Durante a operação do elevador, há basicamente dois eventos a serem considerados:

1. o elevador é chamado em algum andar.
2. o passageiro pressiona o andar para o qual ele deseja ir.

Sempre que o elevador atende um chamado, a entrada do passageiro é sujeita a haver espaço no elevador. O elevador deverá postergar o atendimento do chamado se estiver cheio. Para fins de simplificação, você não precisa considerar os tempos de aceleração e desaceleração do elevador. Mais ainda, você considera que todos os tempos são expressos em uma unidade de tempo Zepsilon. Assim, para percorrer a distância de um andar a outro, é necessário 1 Zepsilon. Para 1 ou mais passageiros entrarem ou saírem do elevador também é necessário 1 Zepsilon.

Em termos da estratégia de escalonamento, ou formas de comandar o elevador, você deverá usar:

- FCFS (First-Come, First-Served). Nesse caso, o elevador deve sempre se mover na direção da primeira requisição (chamada) caso esteja vazio. Se tiver alguém no mesmo, ele deve se mover na direção do primeiro andar acionado no comando.
- SJF (Shortest Job First). Nesse caso, o elevador deverá se mover para o andar mais próximo dentre aqueles acionados/requisitados nos comandos pelos passageiros, caso tenha alguém no mesmo. Caso esteja vazio, o mesmo deve buscar o passageiro mais próximo, dentre os que chamaram o elevador.

Em ambos os caso, o elevador deverá parar ao longo de movimentações se encontrar algum passageiro no caminho e não estiver cheio.

Para cada passageiro transportado, você deve imprimir o tempo de espera e o tempo dentro do elevador até o seu destino, ambos em Zepsilons. Você pode

também imprimir a taxa de ocupação do elevador, assim como a distância percorrida e o número de vezes que a porta foi aberta e fechada.

Assim, elabore um programa com as seguintes características:

1. Projete o seu simulador, incluindo as estruturas de dados a serem utilizadas, assim como a estratégia a ser adotada pelo elevador.
2. O programa deve ser implementado em ao menos 3 módulos. O primeiro módulo deve tratar dos procedimentos de entrada e saída. O segundo módulo contém a lógica do simulador. O terceiro módulo contém o programa principal. Deve ainda ser definido um arquivo de protótipos e definições contendo as estruturas de dados e protótipos das funções dos vários módulos. A compilação do programa deve utilizar o utilitário Make.
3. Todas as estruturas de dados devem ser alocadas dinamicamente, assim como devem ser desalocadas após o processamento.
4. O programa deve receber três parâmetros pela linha de comando, utilizando a primitiva getopt:
  1. arquivo descrevendo o ambiente do elevador
  2. Estratégia de escalonamento <sjf,fcfs>
  3. sequência de eventos a serem tratados.

*A utilização do programa (forma de executar) bem como de compilação utilizando Make devem estar claramente descritas no documento de documentação que acompanhará sua solução (código).*

Por simplicidade, você pode considerar que cada passageiro é um evento, indicando o andar no qual ele chamou o elevador e o andar de destino. Faz parte do trabalho gerar cargas realistas, que, por exemplo, tenham uma distribuição de frequência dos andares desigual indicando um andar de entrada e saída do prédio.

5. Realize várias execuções do seu simulador, mostrando a correção do seu funcionamento.

O programa implementado deve ser avaliado para várias configurações, utilizando as funções `getrusage` e `gettimeofday`. Deve-se também distinguir entre os tempos de computação e tempos de entrada e saída. Comente sobre os tempos de usuário e os tempos de sistema e sua relação com os tempos de relógio. Faz parte do trabalho o desenvolvimento de um programa para criar a carga de trabalho do elevador.

## **Da entrega do trabalho**

A entrega do trabalho deverá ser feita via moodle (aprender) por meio de um único arquivo zip contendo a documentação, entradas utilizadas e código fonte.

## **Avaliação**

O documento a parte “roteiro de documentação” disponível no aprender contém detalhes sobre como fazer uma boa documentação de um trabalho.

### **Deverão ser entregues:**

- listagem das rotinas;
- descrição breve dos algoritmos e das estruturas de dados utilizadas;
- análise da complexidade das rotinas;
- entradas de dados utilizadas para avaliar o algoritmo;
- análise dos resultados obtidos.

### **Distribuição dos pontos:**

- execução  
  execução correta: 40%  
  saída legível: 10%
- estilo de programação  
  código bem estruturado e legível: 15%
- documentação  
  comentários explicativos: 05%  
  análise de complexidade: 10%  
  análise de resultados: 20%