

Primeiro Trabalho de Programação M&S

Esse trabalho foi desenvolvido pelos alunos Breno Corrêa e Salomão Alves. Todo o desenvolvimento foi utilizando a linguagem de programação Python, onde montamos um programa que simula um sistema de atendimento com ou sem fila.

Documentação

1. Introdução

Programa de simulação de sistema de atendimento com ou sem fila, totalmente desenvolvido utilizando python como linguagem de programação.

Seu funcionamento consiste em informações de entrada dadas pelo usuário, e com base nesses dados, simular o funcionamento desse sistema e gerar resultados relevantes para o usuário sobre essa execução.

2. Dados de entrada

Os dados de entrada são informações de configuração que serão utilizadas na simulação, como por exemplo, opções de fila, número de eventos que deseja simular, se o tempo de chegada e o tempo de serviço serão determinísticos ou aleatórios.

Em caso de escolha de valores aleatórios, será necessário o usuário passar alguns valores de simulação para criar uma distribuição de frequência para utilizar o modelo de Monte Carlo. Porém, em caso de escolha de valores determinísticos, durante a execução da simulação será pedido valores para o usuário determinar.

3. Execução

Durante a execução da simulação, os eventos serão mostrados na tela, possibilitando ao usuário acompanhar as chegadas e saídas de clientes, a ocupação da fila, e o tempo de cada evento.

Alguns valores serão mostrados em siglas para facilitar a interface, os significados de cada uma são:

HC	Hora da Chegada
HS	Hora da Saída
TR	Tempo da Simulação
ES	Estado do Servidor
TF	Tamanho da Fila
TS	Tempo de Serviço
TEC	Tempo entre Chegada

4. Relatório

Ao final da simulação, será demonstrado alguns resultados importantes para o usuário, são eles:

- Tempo total de espera na fila
- Número total de clientes

- Número de clientes que esperaram
- Tempo livre do operador
- Tempo total de serviço
- Tempo total no sistema
- Tempo médio de espera da fila
- Probabilidade de um cliente esperar na fila
- Probabilidade do operador livre
- Tempo médio de serviço
- Tempo médio despendido no sistemas

5. Código

Primeiro será executado a *main()*, onde pegará os dados e criará um objeto da classe *service()*, esse objeto será utilizado para salvar as informações importantes durante a execução.

Caso for escolhido gerar os valores de TEC ou TS aleatoriamente, será executado a função *createMCCTable()*, que é responsável por pegar os valores das classes e o número de observações, e assim, gerar a frequência acumulada e salvar num vetor para uso posteriormente.

Depois é chamado a função *simu()*, que será responsável por sincronizar as chegadas e saídas de cliente, dentro de um *for* que repete igual ao número de eventos que o usuário escolheu simular.

Para cada evento de chegada ou saída, é gerado de acordo com a opção que o usuário escolheu (aleatório ou determinístico). Se o valor for determinístico, requisitará o usuário para digitar o valor de sua preferência, em caso de aleatorio, chamará uma função da biblioteca *random* para gerar um número aleatório real entre 0 e 1, com esse número, encontrará o valor da classe utilizando o array de frequência acumulada gerado anteriormente.

Ao final é chamado a função *printStats()* que pega os valores do objeto *s* que é uma instância da classe *service()* (citado anteriormente), com esses valores é calculado os resultados finais da simulação e imprimidos na tela.

6. Bibliotecas

Foram utilizados apenas duas bibliotecas:

random - Para gerar valores aleatórios de 0 a 1

numpy - Para utilizar arrays com mais fácil manipulações de dados