

Engenharia Informática  
Sistema Operativos  
2018 – 2019

**Fase 2: Simulação de uma Loja**



**Docentes:**

Eduardo Marques

Luís Gaspar

**Discentes:**

André Telo, nº 2042316

Lisandro Marote, nº 2030315

Funchal, 14 de Dezembro 2018

## Índice

	Pág.
Introdução .....	3
Funcionamento da loja .....	4
Utilização dos mecanismos de sincronização.....	5
Conclusão.....	8
Anexos .....	9

## Introdução

No âmbito da cadeira de Sistemas Operativos foi-nos proposto a realização de um projeto que visa a implementação de uma simulação de uma loja, onde deverá ter em conta o número de clientes, número de empregados, artigos em stock, artigos em rutura, tempos de espera etc.

Para esta segunda fase pretende-se que o grupo implemente as bibliotecas para comunicação entre o Simulador e o Monitor, incluindo também a interface para o utilizador onde neste deverá mostrar informação pertinente ao utilizador do estado corrente da sua simulação.

O grupo delineou e analisou todos os pormenores presentes no enunciado e esperados para esta segunda fase (sincronização), recorrendo a fluxogramas e esboços com todos os pormenores captados, fazendo chegar este ao professor Luís Gaspar (professor prática laboratorial) dando a sua opinião acerca do plano estipulado pelo grupo se cumpria todos os requisitos pretendidos para esta segunda fase.

## Funcionamento da loja

### Clientes

A loja vai ter uma fila de atendimento, em que as pessoas são atendidas pela ordem de chegada. Os clientes vão ter um tempo de desistência, ao entrar na fila, no mínimo de 2 minutos e no máximo de 10 minutos, sendo que 2 minutos no tempo real são equivalente a 2 segundos na simulação. Os clientes vão ser gerados aleatoriamente, sendo que a cada 10 segundos podem ser gerados 1 a 5 clientes. É também na geração aleatória do cliente que é gerada a intenção do produto que cliente pretende como também se tem intenção de efetuar a troca do produto do ato de atendimento.

### Empregados

A loja inicialmente tem um empregado, cada vez que está 7 clientes na fila é alocado um novo empregado, ou seja, o número de empregados vai ser o resto da divisão do número de clientes na fila. Quando o produto estiver abaixo de uma certa quantidade o empregado a repor será o primeiro que acabar o atendimento ao cliente, tendo os empregados a prioridade de verificar o stock e depois atender os clientes, caso seja um empregado único os clientes esperam que o empregado termine a reposição.

### Produtos

A loja vai conter 3 produtos A, B e C. Cada produto vai conter uma quantidade total de 10 e um tempo de serviço de 5 segundos. Quando chega a uma quantidade menor que 3 o primeiro funcionário a acabar de servir o cliente será o funcionário que irá repor. Quando o funcionário repor o produto, este fica sempre com a quantidade total de 10.

Os produtos serão escolhidos/gerados aleatoriamente pelos clientes. Cada produto vai ter um valor de quantidade cada vez que o cliente pede esse produto o seu valor é decrementado até chegar a um valor em que deve ser repostado. A quantidade do produto pode ser incrementada na rara exceção em que o cliente pede a troca por outro produto.

## Utilização dos mecanismos de sincronização

O projeto vai conter um semáforo para a fila dos clientes que controla o funcionamento da fila sendo que cada cliente só é atendido se houver um funcionário livre (não está a atender nem a repor). Neste não irá haver prioridades entre clientes, simplesmente a ordem de chegada dos clientes irá ser a ordem de atendimento, poderá acontecer alguns clientes desistirem devido ao tempo de espera neste.

Cada empregado vai conter um trinco para garantir a exclusão mútua, sendo que cada empregado atende um cliente de cada vez.

Cada produto vai conter um trinco para que quando um funcionário busque ou reponha o produto nenhum outro funcionário possa buscar o mesmo. Anterior ao trinco, cada produto também irá conter um semáforo de maneira a precaver a situação em que está mais do que 1 empregado a aceder aos mesmos produtos, evitando a hipótese de o produto ficar sem stock, onde caso isto aconteça (irá primeiro soar o alerta que o produto está abaixo da quantidade mínima estipulada) o último empregado presente no semáforo irá ficar responsável pela reposição do mesmo produto, de maneira a dar prioridade ao tempo de espera dos clientes atendidos mais recentemente.

## Políticas de acesso

A comunicação será feita através de sockets. No simulador é escrita a mensagem a enviar para o socket. No monitor é mostrado a mensagem que foi escrita no simulador através do socket.

O monitor recebe uma mensagem:

```
Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja
|                                     Dados da Simulacao                                     |
|.ID_Cliente=30                       |
|.Estado=1                           |
|.Produto=3                          |
|.Desistências=0                     |
|.Empregados=1                       |
```

Código do monitor:

```
int* Buffer=Recebe_Mensagem(sockfd); //(newsockfd);
int ID_Cliente=Buffer[0]; //Numero Processo
int Estado=Buffer[1]; //Ver Tabela de Estados
int produto=Buffer[2]; //Produto a escolher (Produto A,B ou C)
int desistencias=Buffer[3];
int pac_min=Buffer[4];
int pac_max=Buffer[5];
int stock_prod=Buffer[6];
int empregados=Buffer[7];
int cliente_troca=Buffer[8];

switch(Estado) //Evento
{
    case 1: //Simulação

        //Fazer os printf();
        system("clear");
printf("                                \n");
printf("Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja \n");
printf("                                \n");
printf("                                Dados da Simulacao \n");
printf("    .ID_Cliente=%d",ID_Cliente);
printf("\n");
printf("    .Estado=%d",Estado);
printf("\n");
printf("    .Produto=%d",produto);
printf("\n");
printf("    .Desistências=%d",desistencias);
printf("\n");
printf("    .Empregados=%d",empregados);
printf("\n");
printf("                                \n");
printf("                                \n");
```

O simulador envia a mensagem:

```
Pressione space Para Enviar Dados

Mensagem 1
Mensagem 2
Mensagem 3
Mensagem 4
Mensagem 5
Pressione space Para Fechar Simulacao
Enviou Mensagem Fim Simulacao!
```

Código do simulador:

```
printf("Pressione space Para Enviar Dados \n");

char Tecla1=0;
while (Tecla1!=' ') {Tecla1=getchar();}
printf("Mensagem 1 \n");
Escreve_Mensagem(newsockfd,45,1,2,0,2,10,1,1,0);
//0 segundo parametro tem de ser (1), Tem de Ficar Sempre 1 -> Se Estado For =50, Simulacao Acaba.
sleep(4);
printf("Mensagem 2 \n");
Escreve_Mensagem(newsockfd,54,1,1,0,2,10,8,1,0);
sleep(4);
printf("Mensagem 3 \n");
Escreve_Mensagem(newsockfd,2,1,3,0,2,10,6,1,0);
sleep(4);
printf("Mensagem 4 \n");
Escreve_Mensagem(newsockfd,30,1,3,0,2,10,7,1,0);
sleep(4);
printf("Mensagem 5 \n");
Escreve_Mensagem(newsockfd,65,1,2,0,2,10,8,1,1);
```

## Conclusão

Ao longo da elaboração desta fase do projeto verificou-se que o grupo deveria ter-se preparado melhor no sentido de ter começado mais cedo a elaborar o projeto de forma a não deixar acumular para a data limite da entrega, pois com as restantes cadeiras e projetos condicionou de certa forma a progressão do projeto estando simplesmente a cumprir os requisitos mínimos para esta segunda entrega.

Para a terceira fase, o grupo terá que reformular novamente todo o seu plano e melhorar a implementação daquilo que tem estipulado de maneira a garantir o progresso de todo projeto.



## Anexos