



Sistemas Operativos 2022/2023

Relatório Final

Índice

Implementação	3
Frontend	3
Backend.....	5

Implementação

Frontend

```
if(argc != 3)
    Abort("[ERRO] Sintaxe Errada\nSintaxe Correta: ./frontend <USERNAME> <PASSWORD>\n");
```

No início, o frontend verifica se o número de argumentos está correto (se contêm o username e a password respetiva).

```
if(access("BF", F_OK) != 0)
{
    Abort("[ERRO] Servidor fechado...\n");
}
```

O frontend só pode ser lançado se o backend estiver a executar.

Manda a informação para o backend através de um named pipe criado pelo backend.

Se tiver autenticado com sucesso, então cria a sua “caixa postal” onde o backend vai mandar informação relativa a promotores, itens vendidos, etc...

```
read(caixa, &feedback, sizeof(feedback));
if(feedback != 1)
{
    close(caixa);
    close(bf);
    unlink(NomeCaixa);
    if(feedback == 2)
        Abort("Utilizador ja esta online\n");
    else
        Abort("Utilizador nao existe\n");
}
```

Se o username que o utilizador mete estiver online então não deixa entrar.

```
if(getenv("HEARTBEAT") != NULL)
    data.nheartbeat = atoi(getenv("HEARTBEAT"));
else
    data.nheartbeat = 10;
```

O frontend cria 2 threads, a primeira recebe a informação que o backend manda, a segunda manda a cada {HEARTBEAT} segundos (lida das variáveis de ambiente, se não existir, manda a cada 10 segundos), um “tick” a avisar que está vivo.

```
do
{
    fgets(comando, 100, stdin);
    pthread_mutex_lock(&wait);
    if(strcmp(comando, "\n") == 0)
    | continue;
    comando[strcspn(comando, "\n")] = '\0'; //tira "\n" do input;
    if(VerificaComando(comando, &user))
    | write(bf, &user, sizeof(user));
    pthread_mutex_unlock(&wait);
} while(strcmp(comando, "exit") != 0 && data.forceExit != 0);
```

A função principal pede comandos ao utilizador, verifica-os e manda para o backend e quem as processa é a thread 1 mencionada em cima.

Comandos de utilizador (frontend):

- List – Lista todos os itens
 - Licat – Lista todos os itens de uma categoria
 - Lisel – Lista todos os itens de um vendedor
 - Lival – Lista todos os itens com preco até X valor
 - Litime – Listar todos os itens com prazo até X segundos
- Cash – Consultar o saldo
- Add – Carregar o saldo
- Buy – Licita um item
- Sell – Coloca um item a leilão
- Time – Obtém a hora atual
- Exit – encerra o frontend

```
data.continua = 0;
pthread_join(thread[0], NULL);
pthread_join(thread[1], NULL);
pthread_mutex_destroy(&wait);

close(caixa);
close(bf);
unlink(NomeCaixa);
```

Ao terminar, fecha a “caixa postal”, fecha a conexão ao backend e apaga a sua “caixa postal”.

Podemos dizer que o funcionamento do frontend é a seguinte:

1. Manda informação ao backend o seu login. O backend manda de volta o feedback. Dependente do feedback, dá login ou não.
2. Ao entrar, cria um named pipe com o seu PID.
3. Cria threads que recebem a informação que o backend manda.
4. Pede comandos ao utilizador. Durante isto, as threads no entanto, vão recebendo informações sobre promoções, itens vendidos, etc...
5. Se o utilizador sair, o backend tira-o da plataforma. O frontend apaga a sua “caixa postal” e fecha o named pipe do servidor (backend), as threads acabam.

Backend

```
getFileNames(env, filename);  
  
mkfifo("BF", 0666);  
bf = open("BF", O_RDWR);
```

O backend cria um named pipe.

```
if(getenv("HEARTBEAT") != NULL)  
    data.maxheartbeat = atoi(getenv("HEARTBEAT"));  
else  
    data.maxheartbeat = 10;  
  
strcpy(data.userfilename, filename[0]);  
strcpy(data.itemfilename, filename[1]);  
strcpy(data.promfilename, filename[2]);
```

Vai buscar às variáveis de ambiente os nomes dos ficheiros a serem lidos assim como o heartbeat definido.

Depois, cria 2 threads. Uma vai tratar do input do frontend e output para o frontend. A segunda, vai gerir o tempo do backend.

Depois podem existir entre 0-10 threads de promotores lidas do comando "reprom". Cada thread faz um fork com unnamed pipes e lê informação de cada promotor aberto. Só podem existir até 10 promotores.

A função principal trata dos comandos de administrador (visto que o backend é só gerido e lançado pelos administradores).

Comandos de administrador (backend):

- Users – Lista utilizadores online
- List – Lista itens à venda
- Kick – Bane um cliente
- Prom – Lista todos os promotores atualmente ativos
- Reprom – Atualiza promotores
 - Fecha todos os promotores, e lê do ficheiro a nova listagem de promotores. As promoções continuam ativas mesmo depois de fechadas.
- Cancel – Fecha os promotores
 - Fecha todos os promotores com nome X.
- Close – Encerra a plataforma
 - Ao terminar, manda informação que o backend vai terminar ao frontend.

O backend recebe comandos do frontend (já validados) e manda a informação relevante de volta. Sempre que um item é vendido/posto á venda, promoção começa/acaba, o backend envia para o frontend esta informação.

O backend recebe a cada {HEARTBEAT} segundos lidas da variável ambiente, informação do frontend, se não receber, então significa que o frontend saiu sem avisar, retira-o da plataforma.

Os ficheiros de items/utilizadores são atualizados sempre que um novo item é posto á venda, um item é vendido ou o backend é fechado.