

# Sistemas Operativos

MEDICALso

2017019992 - Diogo Semião  
2014010395 - José Valdiviesso

## Implementação

Implementamos a primeira meta do programa *balcão* que utiliza pipes de *stdin* e *stdout* para comunicar com o programa classificador fornecido pelos professores.

Iniciamos o programa por criar os *pipes* e lançamos o programa classificador fazendo um *fork* que se transforma no classificador com recurso à função *execl*. De seguida inicializamos algumas variáveis para ajudar no seguimento do programa. Verificamos se existem as variáveis de ambiente necessárias para o programa correr, e caso não existam avisamos o utilizador e terminamos.

O ciclo principal do programa consiste em pedir ao utilizador os seus sintomas a cada iteração. Verificamos se o utilizador quer sair e caso queira mandamos o classificador terminar e terminamos também o balcão. Caso o utilizador nos tenha dado os seus sintomas vamos enviá-los ao classificador e se seguida lemos a resposta envia pelo mesmo. Separamos a especialidade da prioridade usando um *sscanf* e adicionamos o utente à lista de espera, caso ainda exista espaço.

# Estruturas de dados

```
struct utente{  
    char nome[256];  
    char especialidade[256];  
    int prioridade;  
} typedef Utente;
```

Define as variáveis dos utentes

```
struct especialista{  
    char nome[256];  
    char especialidade[256];  
} typedef Especialista;
```

Define as variáveis dos médicos

```
struct utente_balcao{  
    int pid;  
    char nome[256];  
    char msg[256];  
} typedef U_B;
```

Define a estrutura de dados a usar nos named pipes para a comunicação entre os utentes e o balcão

```
struct balcao_utente{  
    int pid;  
    char msg[256];  
} typedef B_U;
```

Define a estrutura de dados a usar nos named pipes para a comunicação entre o balcão e os utentes.

```
struct medico_balcao{  
    int pid;  
    char nome[256];  
    char especialidade[256];  
} typedef M_B;
```

Define a estrutura de dados a usar nos named pipes para a comunicação entre os médicos e o balcão

```
struct balcao_medico{  
    int pid;  
    char msg[256];  
} typedef B_M;
```

Define a estrutura de dados a usar nos named pipes para a comunicação entre o balcão e os médicos

```
struct utente_medico{  
    int pid;  
    char msg[256];  
} typedef Consulta;
```

Define a estrutura de dados a usar nos named pipes para a comunicação durante as consultas, entre médico e utente