



Sistemas Operativos 21/22

## **Trabalho Prático – Programação em C para UNIX**

Docente: João Durães

Ano Letivo 2021/2022

**Trabalho feito por:**

Francisco Simões – 2019133920

Pedro Praça - 2020130980

## Estrutura de dados

Cliente.h:

```
typedef struct CLIENTE cliente,*client;
struct CLIENTE{
    pid_t pid_cliente; // pid de cada instância cliente
    char nome_cliente[BUFFER_CLIENTE]; // nome de cada cliente
    char sintomas_cliente[BUFFER_CLIENTE]; // sintomas de cada cliente
    char prioridade_cliente; // prioridade obtida de cada cliente pós classificador
    char especialidade_cliente[BUFFER_CLIENTE]; // especialidade obtida de cada cliente pós classificador
};
```

pid\_t pid\_cliente -> armazena o pid de cada cliente;

char nome\_cliente[BUFFER\_CLIENTE] -> representa o nome de cada cliente;

char sintomas\_cliente[BUFFER\_CLIENTE] -> representa os sintomas introduzidos pelo cliente;

char prioridade\_cliente -> prioridade devolvida pelo classificador;

char especialidade\_cliente[BUFFER\_CLIENTE] -> especialidade devolvida pelo classificador.

Medico.h:

```
typedef struct MEDICO medico,*doctor;
struct MEDICO{
    pid_t pid_medico; // pid de uma dada x instância médico
    char nome_medico[BUFFER_MEDICO]; // nome de uma dada x instância médico
    char especialidade_medico[BUFFER_MEDICO]; // especialidade de uma dada x instância médico
    cliente pid_cliente; // pid do cliente que irá atender
};
```

pid\_t pid\_medico -> armazena o pid de cada médico

char nome\_medico[BUFFER\_MEDICO] -> representa o nome de cada médico

char especialidade\_medico[BUFFER\_MEDICO] -> representa a especialidade de cada médico

cliente pid\_cliente -> representa o cliente que está a ser atendido por esse médico

## Funções implementadas:

Balcao.c:

```
void levariaveisambiente(int *MAXCLIENTES, int *MAXMEDICOS)
```

Função que lê as variáveis de ambiente “MAXCLIENTES” e “MAXMEDICOS” e caso não estejam definidas terá valores por omissão.

## Classifica sintomas:

```
int classificasintomas()
```

Função que cria dois *pipes*, um *pipe* só com uma extremidade aberta, de leitura, c\_b e um *pipe*, só com uma extremidade aberta, de escrita, b\_c. Isto é, o primeiro pipe é usado para o classificador escrever as mensagens para o balcão e por outro lado o segundo pipe, b\_c, serve para o balcão enviar mensagens para o classificador. De seguida cria um processo filho, onde se realiza o redireccionamento do stdin e stdout e posteriormente é lançado o classificador para comunicar através destes pipes, com o balcão. Este envia os sintomas de um dado cliente e o classificador retornará, pós interpretação dos sintomas, a especialidade e prioridade desse mesmo cliente.

## Vetor dinâmico:

cliente\_espera -> corresponde ao armazenamento dos clientes com instâncias ativas sendo que MAXCLIENTES (N) é o número de instâncias máxima por cada execução do balcão;

especialista\_espera -> corresponde ao armazenamento dos médicos com instâncias ativas sendo que MAXMEDICOS (M) é o número de instâncias máxima por cada execução do balcão;

```
client cliente_espera; // armazena os clientes em espe  
doctor especialista_espera; // armazena os medicos em  
  
cliente_espera = malloc( _Size: sizeof (cliente)*N); //AL  
especialista_espera = malloc( _Size: sizeof (medico)*M);
```