

# Sistemas Operativos

## Trabalho Prático

### Argus: Controlo e Monitorização de Processos e Comunicação

Grupo de Sistemas Distribuídos  
Universidade do Minho

13 de Dezembro de 2019

#### Informações gerais

- Cada grupo deve ser constituído por até 3 elementos;
- O trabalho deve ser entregue até às 23:59 de 8 de Janeiro de 2019;
- Deve ser entregue o código fonte e um relatório de até 6 páginas (A4, 11pt) no formato PDF (excluindo eventuais capas e anexos).
- A apresentação do trabalho ocorrerá a 10 de Janeiro de 2019.

#### Resumo

Implemente um serviço de monitorização de execução e de comunicação entre processos. O serviço deverá permitir a um utilizador a submissão de sucessivas tarefas, cada uma delas sendo uma sequência de comandos encadeados por pipes anónimos. Além de iniciar a execução das tarefas, o serviço deverá ser capaz de identificar as tarefas em execução, bem como a conclusão da sua execução. O serviço deverá terminar tarefas em execução, caso não se verifique qualquer comunicação através de pipes anónimos ao fim de um tempo especificado. O serviço deverá ainda terminar as tarefas em execução caso seja especificado um tempo máximo de execução. O serviço a implementar deverá ser constituído por um servidor e um por cliente, que deverão comunicar por pipes com nome. Considere que o servidor necessitará de apenas atender um cliente de cada vez. O cliente deverá oferecer ao utilizador não só uma interface de linha de comando como também uma interface textual interpretada (shell).

#### Funcionalidade

Este serviço deverá suportar, pelo menos, as seguintes funcionalidades:

- definir o tempo máximo (segundos) de inactividade de comunicação num pipe anónimo (opção `-i n` da linha de comando)

```
argus$ tempo-inactividade 10
```

- definir o tempo máximo (segundos) de execução de uma tarefa (opção `-m n` da linha de comandos)

```
argus$ tempo-execucao 20
```

- executar uma tarefa (opção `-e p1 | p2 ... | pn` da linha de comando)

```
argus$ executar cut -f7 -d: /etc/passwd | uniq | wc -l
nova tarefa #5
```

- listar tarefas em execução (opção `-l` da linha de comando)

```
argus$ listar
#1: ./a.out | teste
#5: cut -f7 -d: /etc/passwd | uniq | wc -l
```

- terminar uma tarefa em execução (opção `-t n`)

```
argus$ terminar 1
```

- listar registo histórico de tarefas terminadas (opção `-r`)

```
argus$ historico
#2, concluida: ./a.out | ./b.out
#3, max inactividade: prog1 | prog2 | prog3
#4, max execução: prog2 | prog3
```

- apresentar ajuda à sua utilização (opção `-h`)

```
argus$ ajuda
tempo-inactividade segs
tempo-execucao segs
executar p1 | p2 ... | pn
...
```

## Cliente e Servidor

Deverá ser desenvolvido um cliente que ofereça uma interface com o utilizador via linha de comando que permita suportar a funcionalidade descrita e ilustrada acima. O utilizador poderá agir sobre o servidor através dos argumentos passados na linha de comando do cliente, ou, no caso do cliente ser invocado sem argumentos, através de uma linguagem textual interpretada (shell). Deverá ser também ser desenvolvido um servidor, mantendo em memória a informação relevante para suportar a funcionalidade acima descrita.

Tanto o cliente como o servidor deverão ser escritos em C e comunicar via *pipes com nome*. Na realização deste projecto não deverão ser usadas funções da biblioteca de C para operações sobre ficheiros, salvo para impressão no *standard output*. Da mesma forma não se poderá recorrer à execução de comandos directa ou indirectamente através do interpretador de comandos (p. ex.: *bash* ou `system()`).

Por último, note que – tal como no exemplos apresentados acima – poderão existir múltiplas tarefas em execução num dado momento.