

Enunciado do Trabalho Prático

Este trabalho prático tem como objetivo a aplicação prática dos conceitos de gestão de processos e de ficheiros, assim como a aplicação da comunicação entre processos.

O trabalho deverá ser realizado em grupos de 3 elementos, excepcionalmente com menos elementos, devidamente justificado ao docente.

Parte 1) Implementação um conjunto de comandos para manipulação de ficheiros (9 vals)

Implemente os seguintes comandos através de funções de chamada ao sistema (*system calls*) (ver <https://linux.die.net/man/>). Estes comandos devem ser implementados em C (para Linux) e serão invocados através de um interpretador de comandos. Qualquer mensagem de erro deverá ser apresentada no descritor *stderr*, não utilize a aplicação *shell* ou comandos pré-existentes para executar a funcionalidade pretendida.

- a) **mostra** ficheiro – Este comando deve apresentar no ecrã o conteúdo do ficheiro indicado como parâmetro. Caso o ficheiro não exista (na diretoria de trabalho atual), o interpretador deve avisar o utilizador que o ficheiro não existe;
- b) **conta** ficheiro – Este comando deve contar o número de caracteres existentes de um ficheiro. Se o ficheiro não existir, deverá ser indicado ao utilizador uma mensagem de erro;
- c) **apaga** ficheiro – Este comando deve apagar o ficheiro com o nome indicado (da diretoria atual). No caso de o ficheiro indicado **não** existir, e apenas, deve ser apresentado um aviso ao utilizador;
- d) **informa** ficheiro – Este comando apresenta **apenas** a informação do sistema de ficheiros em relação ao ficheiro dado, tipo de ficheiro (normal, diretoria, link, etc.), i-node, utilizador dono em formato textual;
- e) **acrescenta** origem destino – Este comando deve acrescentar o conteúdo da “origem” no final do “destino”. Caso algum dos ficheiros não exista, deve ser apresentado um aviso ao utilizador;
- f) **lista** [caminho] – Este comando deve apresentar uma lista de todas as pastas e ficheiros existentes no caminho indicado ou na diretoria atual se não especificado. Adicionalmente, deve distinguir ficheiros simples de diretorias através de uma indicação textual.

Parte 2) Implementação de um interpretador de comandos (5 vals)

No sentido de substituir o interpretador de comandos habitual, Bash shell, por um novo interpretador personalizado, deve ser implementada uma aplicação cuja função será a leitura de uma sequência de caracteres da consola, e em seguida executar essa sequência como um comando e respetivos argumentos no sistema. O programa deve mostrar o símbolo “%” como indicação de que está pronto para ler um novo comando do utilizador.

O programa deve executar o comando através de primitivas de execução genérica de processos tendo como referência a funcionalidade da função *system(3)*, mas **sem fazer uso da mesma**. Cada comando deve dar origem a um novo processo. Adicionalmente, pode considerar que a execução do interpretador deve ser suspensa até o comando indicado estar concluído. O interpretador deve indicar sempre se o comando concluiu com ou sem sucesso, através do seu código de erro/terminação. O programa deve permitir executar vários comandos sequencialmente, isto é, um a seguir ao outro, até o utilizador indicar o comando especial “termina” que termina esta aplicação.

```
$ ./interpretador
% lista /home/user/Desktop
...
Terminou comando lista com código 0
% apaga /home/user/Desktop/file
...
Terminou comando apaga com código 1
% termina
$
```

Parte 3) Sistemas Distribuídos - Análise de cópia de ficheiros entre máquinas virtuais (6 vals)

- a) Configure a sua máquina virtual, de modo a que consiga comunicar com o host físico (máquina real). Apresente o resultado do comando *ping* e detalhe a configuração de rede da máquina virtual, bem como a configuração IP de ambas as máquinas usando o comando “ip address show”.
- b) Recorrendo ao comando *iperf3* mostre as diferenças de transferências entre a máquina real e virtual, usando *tcp* e *udp*. Explique os resultados obtidos.
- c) Instale um servidor web (nginx ou apache) e mostre os resultados de um teste de carga ao “url /”, simulando 10 clientes em simultâneo e 500 pedidos cada um, utilizando o utilitário *ab* ou *jmeter*, sendo que o *jmeter* terá um acréscimo de 0.3 valores. Explique os resultados obtidos.
- d) Tendo como base o problema de copiar um ficheiro da primeira máquina virtual para a segunda máquina virtual (este ficheiro deverá ser criado na directoria /home/<nomeutilizador>/, com o nome sosd.txt e com o conteúdo “este ficheiro é para copiar”), indique os comandos que permitem realizar esta operação.

Faça uso das seguintes sugestões de comandos para apresentar até 3 possíveis soluções distintas:

- 1) scp
- 2) dd, nc, pipe (|)
- 3) cat, ssh, pipe (|)

Apresente o comando para cada uma das três possíveis soluções com uma descrição de cada uma delas, indicando qual a melhor em termos de utilização de recursos e rapidez.

Listagem de Chamadas ao Sistema

As chamadas ao sistema que poderão ser utilizadas neste trabalho encontram-se na secção 2 das páginas do Manual de Unix¹.

Gestão de Ficheiros

- open
- read
- write
- close
- dup e dup2
- pipe
- stat(2)
- opendir(3)
- readdir(3)
- closedir(3)

Gestão de Processos

- fork
- execve(2)
- exec(3)
- wait

Gestão de Sockets

- socket(2)
- connect (2)
- accept(2)
- send(2)
- recv(2)

As seguintes funções **não** devem ser usadas na resolução deste trabalho:

- system, fopen, fclose, fread, fwrite, fseek

¹ <https://linux.die.net/man/2/>

Entrega Final

O grupo deverá entregar um arquivo **ZIP**, via Moodle, com a identificação dos membros do grupo e com os ficheiros solicitados. Dentro do arquivo deverão estar os seguintes ficheiros:

- relatório (PDF) documentando a resposta a cada alínea deste enunciado;
- código fonte usado para responder às partes do enunciado.

É exigido que cada relatório contenha no seu início as alíneas em que cada elemento do grupo contribuiu.