Lic. em Eng. de Sistemas Informáticos

Sistemas Operativos e Sistemas Distribuídos

2018/2019

Enunciado do Trabalho Prático para Prova por Experiência Profissional

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

Parte 1) Implementação um conjunto de comandos para manipulação de ficheiros (12 vals)

Implemente os seguintes comandos através de funções de chamada ao sistema (*system calls*) (ver http://manpages.ubuntu.com/manpages/). Cada um destes comandos deve ser implementado em C (para Linux) e funcionar como uma aplicação independente. 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 linhas de um ficheiro. O número de linhas é definido em Unix como sendo o número de caracteres '\n' existentes no conteúdo do ficheiro. Se o ficheiro não existir, deverá ser indicado ao utilizador uma mensagem de erro;

c) **acrescenta** ficheiro1 ficheiro2 – Este comando deve acrescentar o conteúdo do ficheiro2 no final do ficheiro1. Caso algum dos ficheiros não exista, deve ser apresentado um aviso ao utilizador;

d) **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.

Parte 2) Implementação de um interpretador de comandos (4 vals)

Desenvolver um programa que tem a mesma funcionalidade de um interpretador de comandos. Isto é, recebe uma sequência de texto do teclado, e usando funções do sistema operativo, executa esse texto como um comando no sistema.

A aplicação deve fazer a leitura de uma sequência de caracteres da consola, e em seguida executar essa sequência como um comando e respectivos 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 escrever o comando especial "termina" que termina esta aplicação.

```
$ ./interpretador
% ls -l /home
...
Terminou comando ls com código 0
% rm /home/user/Desktop/file
...
Terminou comando apaga com código 1
% termina
$
```

Parte 3) Servidor de entrega de ficheiros usando Sockets TCP/IP (4 vals)

Pretende-se criar uma aplicação que funcionando segundo o modelo cliente-servidor permita ao cliente realizar a operação de fazer o *download* de um ficheiro do servidor para o cliente com base no seu nome. Para executar este comando o cliente deverá enviar uma mensagem para o servidor com o conteúdo: "GET *nome_ficheiro*". O servidor responde com o conteúdo do ficheiro.

Para implementar este programa deve usar apenas as primitivas do sistema, nomeadamente funções que manipulam *sockets* e ficheiros. Não é permitida a utilização de bibliotecas auxiliares que implementem componentes de um servidor TCP ou HTTP. A linguagem de programação para este exercício deve ser C.

O programa servidor deve permitir especificar na linha de comandos qual o porto TCP onde estará à escuta de clientes.

Só é necessário desenvolver uma aplicação servidor. A aplicação cliente pode ser própria, ou genérica como o *telnet*.

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 Unix[1].

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

Entrega Final

A entrega deve ser feita num arquivo **ZIP**, com os seguintes ficheiros:

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

[1] http://manpages.ubuntu.com/manpages/
http://manpages.ubuntu.com/manpages/bionic/pt/man2/
http://manpages.ubuntu.com/manpages/bionic/en/man2/socket.2.html