Escola Superior de Tecnologia e Gestão de Beja
Curso de Engenharia Informática

Sistemas Operativos

Trabalho de Grupo N.º 2

Shell Simples para o Linux

Luís Garcia

Neste trabalho de grupo pretende-se que os alunos desenvolvam uma shell simples denominada ***ss (simple shell)***. A ***ss*** deve aceitar e executar comandos introduzidos pelo utilizador. Cada comando deve ser executado por uma *shell* filha. Como o número de argumentos de cada comando é variável, a *shell* filha deve utilizar a função ***execvp*** para a execução do comando. A *shell* deve esperar o término da *shell* filha antes de aceitar um novo comando. Este trabalho está dividido em duas partes. Na primeira parte é solicitada em cada alínea o desenvolvimento de uma função que deverá ser utilizada na solução base. Na segunda parte é solicitado que os alunos continuem de forma autónoma a implementação de outras funcionalidades mais avançadas da *shell*. A seguir é apresentada uma possível utilização de uma versão base desta *shell*.

```
[lfnhbg@localhost lab3]$ ss

Introduza um comando:ls
fich1.txt  ss  ss.c

Introduza um comando:cat fich1.txt
Este ficheiro serve
para uma experiencia

Introduza um comando:cp fich1.txt fich2.txt

Introduza um comando:ls
fich1.txt  fich2.txt  ss  ss.c

Introduza um comando:cat fich2.txt
Este ficheiro serve
para uma experiencia

Introduza um comando:quit

[lfnhbg@localhost lab3]$
```

(14 valores)

Parte 1

a) Desenvolva a função ***void GetCommand(char command[])***. Esta função deve aceitar um comando do utilizador e devolvê-lo através do parâmetro ***command***. Experimente esta função dentro de um ciclo que deverá ser interrompido quando o utilizador introduzir o comando ***quit***.

b) Desenvolva a função ***int PrintArguments(char command[])***. Esta função deve escrever cada um dos argumentos do comando numa linha diferente do ecrã. Para tal utilize a função ***strtok***. Para testar o funcionamento da função chame-a sempre que o utilizador introduza um novo comando.

c) Com base na função da alínea anterior desenvolva a função void ***makeArgVector(char command[], char *argVector[])***. Esta função deve construir um vector que contenha em cada posição um dos argumentos do comando. Na posição seguinte ao último argumento deve ser colocada a constante NULL como indicação do fim dos argumentos. As posições do vector devem ser alocadas de forma dinâmica através da função ***malloc***.

d) Conclua o desenvolvimento desta versão base da ***ss***. Para tal a *shell* deve aceitar o comando e em seguida lançar uma shell filha. A *shell* filha deve construir o vector de argumentos e executar o comando através da função ***execvp***. A *shell* deve esperar pelo término da *shell* filha para então ficar disponível para a introdução de um outro comando. Este processo deve repetir-se até o utilizador introduzir o comando ***quit***.

(6 valores)

Parte 2

Nesta parte do trabalho pretende-se que os alunos sigam a lógica de programação proposta na parte 1, passo-a-passo, para desenvolverem funcionalidades mais avançadas na *shell*. As funcionalidades que os alunos devem tentar implementar são as seguintes:

- implementar a possibilidade de execução de listas de comandos, por exemplo: *sleep 5; date; sleep5; date*
- implementar a possibilidade de utilização da redireção, por exemplo: *ls -l > file_list.txt*
- implementar a possibilidade de utilização de pipelines, por exemplo: *ls -l | grep '^d' | wc -l*

No desenvolvimento deste trabalho deverão ser utilizadas essencialmente as bibliotecas e funções abordadas nas aulas.

Avaliação

Os trabalhos serão avaliados de acordo com a quantidade e qualidade das funcionalidades implementadas. Apenas serão aceites trabalhos desenvolvidos parcialmente nas aulas da disciplina. Não são aceites soluções desenvolvidas exclusivamente fora das aulas. Os vários elementos do grupo devem participar na elaboração do trabalho. **Não serão toleradas cópias nas soluções apresentadas. Nestas situações os alunos obterão nota zero no trabalho.**

Grupos de Trabalho

O trabalho deve ser desenvolvido por grupos com um máximo de três elementos. Não são permitidas alterações nos elementos do grupo salvo em situações extraordinárias e devidamente justificadas.

Entrega do Trabalho

Os alunos devem entregar o ficheiro executável com o programa, um pequeno relatório que apresente a solução encontrada e a apresentação electrónica do trabalho. No relatório devem ser apresentadas e explicadas as principais partes do código. Num anexo deste relatório deve ser colocada a totalidade do código da aplicação. Para a entrega do trabalho os alunos devem compactar o ficheiro executável, relatório e apresentação num único ficheiro (zip ou equivalente). O nome deste ficheiro deve conter a indicação TG1 (Trabalho de Grupo 1) e o número dos alunos que compõem o grupo. Por exemplo TG1_2000_3000_4000.zip seria o Trabalho de Grupo 1 do grupo formado pelos alunos com os números 2000, 3000, e 4000. O trabalho deve ser entregue através do sistema *Moodle*. **Não serão consideradas soluções entregues por e-mail.**

Apresentação do Trabalho

Os alunos deverão apresentar de forma individual a solução desenvolvida. Nesta apresentação os alunos devem demonstrar conhecer a totalidade do trabalho e estar aptos a realizar modificações ao código apresentado de acordo com as indicações do

docente. Esta apresentação será realizada num horário especialmente marcado para o efeito.

Bom Trabalho
Luís Garcia