

Sistemas Operativos, 3.º MIEIC 2009/10, FEUP

– RMA, JVV, HSF –

October 12, 2009

1.º Trabalho Prático: SO Shell (*sosh*)

Objectivos

Conhecer e utilizar a interface de desenvolvimento de sistemas baseados em Unix. Nomeadamente, após a conclusão deste trabalho com sucesso, os alunos serão capazes de conseguir

- criar novos processos;
- fazê-los intercomunicar por sinais;
- percorrer um sistema de ficheiros e dele obter informações.

Descrição

Uma shell é uma aplicação que lê os comandos introduzidos pelo utilizador numa consola e os executa de acordo com uma semântica própria, como por exemplo bash (<http://www.gnu.org/software/bash/bash.html>) e ksh (<http://www.kornshell.com/>). O trabalho prático consiste em desenvolver um interpretador de comandos muito simples chamado *sosh*. A construção do interpretador será feita por etapas (sugestão: crie cópias de segurança sucessivas entre versões!).

1. Implemente a *sosh0.1* suportando os seguintes requisitos:
 - está em loop a servir comandos introduzidos pelo utilizador;
 - reconhece o comando *quem* que serve executando o comando Unix *who* num processo independente;
 - o processo pai deve esperar pelo processo filho;
 - ao receber o sinal Ctrl-C, a aplicação pergunta ao utilizador se pretende terminar (S/N).
2. Implemente a *sosh0.2* suportando os seguintes requisitos:

- *psu* - lista os processos em execução do utilizador actual.
- *ver* - imprime no *stdout* a versão do sosh (e.g., “sosh versão 0.2”).
- *ajuda* - lista os comandos suportados.
- *localiza <cmd>* - localiza os caminhos (absolutos) no qual o comando *cmd* aparece (compare com *locate <cmd>*, implementação deve ser independente do *locate*, usando primitivas oferecidas pela linguagem).
- *exit* - sai da shell.

Sugestões:

- Se utilizar a versão *exerc1p*, não é necessário indicar o caminho absoluto do comando desde que este se encontre na variável de ambiente *PATH*.
- Utilize o *gdb* como ajuda para identificar se a leitura e parsing dos comandos de entrada são feitos correctamente.

3. Implemente a sosh0.3 suportando os seguintes requisitos:

- Além dos comandos anteriores, a nova versão deve permitir executar as aplicações localizadas em */usr/bin*, com número variável de argumentos. Sugestões:
 - Faça parsing da linha de comandos utilizando o *strtok* (man *strtok*).
 - Utilize a variante *execv* para suportar argumentos variados (man *execv*).
- Organize o código da aplicação colocando a implementação dos comandos da sosh num módulo (.h e .c) independente:
 - Adapte o ficheiro *cmds.h* (ver Anexo A) e coloque a implementação respectiva no ficheiro *cmds.c*.
 - O programa principal deve encontrar-se no ficheiro *sosh.c*.
 - Adapte a Makefile no Anexo B para suportar regras implícitas.

Valorização

Quem desejar obter o máximo da classificação estabelecida para este trabalho poderá estender sosh0.3 (versão b) por forma que possa suportar as seguintes opções

- *hist* - lista todos os comandos já executados ordenadamente precedidos pelo o número de ordem respectivo
- *!#* - executa o mais recente comando executado no sosh começado pela string *#*. A string *#* pode ter uma dimensão até 32 caracteres (ex. “!*l*” equivale a executar o comando “*ls -l*”, assumindo que esta foi o última chamada ao comando “*ls*” executada no sosh)

Sugestões:

- Utilize uma lista dinâmica para armazenar strings (pode modificar a lista de inteiros da primeira aula).
- Adicione o código fonte da lista como ficheiros independentes (.h e .c) e actualize a Makefile.

Producto final

O trabalho total consiste na produção de ficheiro compacto incluindo o código fonte, um makefile preparado para facilitar a geração do executável, e um ficheiro de texto README com os passos necessários para compilar e executar o código. Terão também de entregar um breve relatório (não listem a totalidade do código fonte no documento - apenas partes relevantes) em formato pdf. O compacto, identificado com um nome do tipo TPnGX.ext, onde n é o número do turno prático que estão inscritos e X é o número atribuído ao grupo pelo docente das teórico-práticas (e.g. TP1G2.tgz), deve ser entregue usando as facilidades disponíveis para a unidade curricular no moodle (<http://moodle.fe.up.pt/0910/>) até ao final do dia **01/11/2009**.

Avaliação

A classificação do trabalho será repartida em 90% para o programa normal e 10% para a parte de valorização. O programa normal será classificado relativamente aos seguintes aspectos:

- **programação**, incluindo estrutura, comentários e funcionalidades (60%); sendo que a valorização valerá 3 valores e o restante 17 valores.
- **relatório curto**, que deve consistir na apresentação dos objectivos, estrutura trabalho, testes, conclusão e declaração de autoria (20%);
- **apresentação**, brevemente efectuada pelos autores ao docente que apreciará, ele mesmo, na altura, as facilidades de compilação, utilização, etc. (20%).

Código de Honra

Espera-se que cada grupo obtenha a sua nota baseada exclusivamente no esforço e trabalho dos elementos que constituem o grupo. Consequentemente qualquer forma de plágio, constitui uma fraude inaceitável. Estes comportamentos não serão tolerados e os alunos serão punidos de acordo com o grau de severidade de acordo com a regulamentação da FEUP.

Bom trabalho!
RMA, JVV, HSF.

A cmds.h

```
1 /*
   * cmds.h - sosh command module
3 */

5 #ifndef _CMDS.H_
   #define _CMDS.H_
7
9   int cmd_quem();
11  int cmd_psu();
13  int cmd_usrbin(char **argv, int argc);
15  int cmd_exit();
17  int cmd_localiza();
19  int cmd_ver();
21  int cmd_ajuda();
   #endif
```

B Makefile

```
# Makefile
#
# Implicit Rules
#
#   Implicit rules are predefined commands for predefined file types, such
#   as .c, .o and others.
#
#   Implicit rules relieve programmers from having to specify all the
#   commands. Typically, implicit rules are parameterized by macros such as
#   the CC and CFLAG macros.
#
#   Since we are using implicit rules, it is possible to omit the commands
#   out of the rules below

CC = gcc
CFLAGS = -g

sosh: sosh.o cmds.o
# gcc -o sosh sosh.o cmds.o

sosh.o: sosh.c cmds.h
# gcc -g -c sosh.c
```

```
cmds.o: cmds.c
# gcc -g -c cmds.c

clean:
rm -f *.o sosh
```