/\* ================================================================== \*

Universidade Federal de Sao Carlos - UFSCar, Sorocaba

Disciplina: Laboratorio de Sistemas Operacionais

Prof. Gustavo Maciel Dias Vieira

Projeto 2 - Gerencia de Projetos

Descricao: interpretador de comandos basico.

Daniel Ramos Miola RA: 438340

Giulianno Raphael Sbrugnera RA: 408093

\* ================================================================== \*/

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/wait.h>

#define MAX 256

int main()

{

char linhaComando[MAX];

char \*token;

char \*\*arg;

int pid;

int tam, n;

int wait = 0;

char \*in, \*out;

while (1)

{

//prompt de comando

printf("> ");

fgets(linhaComando, 256, stdin);

tam = strlen(linhaComando);

//verifica tamanho, ignora se estiver vazia

if(tam==1)

continue;

//substitui \n no final da string

linhaComando[tam - 1] = '\0';

//zera variaveis de verificação de wait e arquivos de entrada e saida

wait = 0;

in = NULL;

out = NULL;

//aloca e retira comando

token = strtok(linhaComando," ");

arg = (char\*\*)malloc((sizeof(char\*))\*20);

arg[0] = token;

//separa argumentos retirando redirecionamentos e wait

n = 0;

while (token != NULL && n <= 19)

{

n++;

token = strtok(NULL, " ");

if(token == NULL)

break;

if(!strcmp(token,"<")){

token = strtok(NULL," ");

in = token;

n--;

}else if(!strcmp(token,">")){

token = strtok(NULL," ");

out = token;

n--;

}else if(!strcmp(token,"&")){

wait = 1;

n--;

}else{

arg[n] = token;

}

}

//seta ultimo argumento nulo

arg[n+1] = NULL;

if (!strcmp(arg[0], "exit"))

{

exit(EXIT\_SUCCESS);

}

pid = fork();

if (pid)

{

if(!wait)

waitpid(pid, NULL, 0);

}

else

{

//verifica e faz redirecionamentos

if(in!=NULL)

freopen(in,"r",stdin);

if(out!=NULL)

freopen(out,"w",stdout);

execvp(arg[0],arg);

if(in!=NULL)

fclose(stdin);

if(out!=NULL)

fclose(stdout);

printf("Erro ao executar comando!\n");

exit(EXIT\_FAILURE);

}

}

}