

Autores: Anselmo Soethe Nurnberg Junior
Ramon Dutra Miranda

Descrição:

No arquivo Lab3a_ansejr_ramon.rdm.c implementamos um micro-shell que pode reconhecer comandos básicos do Unix. Utilizamos o fork() para criar um filho do processo shell e assim executar seus comandos.

Para compilar utilizamos o gcc da seguinte maneira: "gcc -o microshell Lab3a_ansejr_ramon.rdm.c" e o executamos "./microshell", possibilitando assim a utilização até o usuário digitar o comando "exit" que fará o microshell se encerrar.

No arquivo Lab3b_ansejr_ramon.rdm.c construímos um programa que após sua execução terá um determinado tempo para que o usuário digite alguma coisa, se digitar ele terá o mesmo tempo para digitar outra e assim sucessivamente, caso não tenha entrada o programa é encerrado por timeout. A compilação é feita usando o gcc da seguinte maneira: "gcc -o alarme Lab3b_ansejr_ramon.rdm.c". Já para sua execução fazemos "./alarme 5" onde 5 é o número de segundos que o programa espera por uma entrada do usuário, caso não seja passado argumento ou o mesmo for 0 (zero) o programa é terminado logo após o início de sua execução.

Codigos:

```
//3.1
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

char input[1024];
char pasta[4096];
char *argv[3];

int main(void) {
    while(1) {
        printf("mshell> ");
        fgets(input, 1024, stdin);
        input[strlen(input)-1] = '\0';
        if(!strcmp(input, "exit"))
            exit(0);

        argv[0] = strtok(input, " ");
        argv[1] = strtok(NULL, " ");
        argv[2] = NULL;

        if(!strcmp(argv[0], "cd")){
            if(chdir(argv[1])!=0)
                printf("caminho inválido\n");
        }else{
            if(fork() == 0){
                if(execvp(argv[0], argv) == -1){
                    printf("comando inválido\n");
                }
            }
        }
    }
}
```

```

        exit(1);
    }
} else
    wait(NULL);

}
}
}

```

// 3.2

```

#include <signal.h>
#include <stdio.h>
#include <unistd.h>

void timeout_function()
{
    printf("Tempo esgotado (timeout)!\n");
    exit(1);
}

int main(int argc, char* argv[]){
    char a[100];
    int tempo;

    if(argv[1]==NULL)
        timeout_function();
    tempo = atoi(argv[1]);
    if(tempo == 0)
        timeout_function();
    signal(SIGALRM, timeout_function);
    while(1) {
        alarm(tempo);
        printf("Digite uma entrada: \n");
        scanf("%s", &a);
        alarm(0);

    }

    return 0;
}

```