

LICENCIATURA EM ENGENHARIA DE SISTEMAS E INFORMÁTICA
LICENCIATURA EM MATEMÁTICA E CIÊNCIAS DA COMPUTAÇÃO

Sistemas Operativos I

Mini-teste de avaliação¹

21 de Novembro de 2005

Duração: 1h00m

Protótipos de algumas funções e chamadas ao sistema relevantes

Processos

- `pid_t fork(void);`
- `void exit(int status);`
- `pid_t wait(int *status);`
- `pid_t waitpid(pid_t pid, int *status, int options);`
- `WIFEXITED(status);`
- `WEXITSTATUS(status);`
- `int execlp(const char *file, const char *arg, ...);`

- `int execvp(const char *file, char *const argv[]);`
- `int execve(const char *file, char *const argv[], char *const envp[]);`

Sinais

- `void (*signal(int signum, void (*handler)(int)))(int);`
 - `int kill(pid_t pid, int signum);`
 - `int alarm(int seconds);`
 - `int pause(void);`
-

I

1 Descreva os estados por que um processo pode passar durante o seu tempo de vida, isto é, desde o momento da sua criação até ao momento sua terminação).

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

II

Analise, corrija e complete o código da página seguinte com base no seguinte enunciado:

Pretende-se monitorizar (todos os minutos) o custo de comunicação em função de um tarifário apresentado em baixo. O estabelecimento e o fim da ligação são assinalados através de SIGUSR1 e SIGUSR2 respectivamente. Apresente o programa **monitor** recorrendo às primitivas de sinais do UNIX SYSTEM V.

Duração	Custo
$t < 5min.$	5 cent./min.
$t \geq 5min.$	4 cent./min.

```

1  /* includes: stdio.h, stdlib.h, unistd.h, signal.h */
2
3  volatile int flag;
4
5
6
7
8  int handler1(int signum)
9  {
10
11
12
13
14
15
16  }
17
18  void handler2(int signum)
19  {
20
21
22
23
24
25
26  }
27
28  int handler3(int signum)
29  {
30
31
32
33
34
35
36  }
37
38  int main(int argc, char **argv)
39  {
40
41
42      if (signal(SIGUSR1, handler1) == SIG_ERR) {
43          perror("signal");
44          exit(1);
45      }
46      if (signal(handler2, SIGUSR2) == SIG_IGN) {
47          perror("signal");
48          exit(2);
49      }
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66      while ( ) {
67
68          pause();
69
70      }
71
72      return 0;
73  }
74

```