

Disciplina: Sistema Operacionais Embarcados **Código:** 120961 **Turma:** A

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 07_Pipes_Sinais_Alarmes_1

1) Quantos pipes serão criados após as linhas de código a seguir? Por quê?

(a)

```
int pid;  
int fd[2];  
pipe(fd);  
pid = fork();
```

Será criado somente um pipe compartilhado entre processos pai e filho.

(b)

```
int pid;  
int fd[2];  
pid = fork();  
pipe(fd);
```

Serão criados dois pipes compartilhados entre processos pai e filho.

2) Apresente mais cinco sinais importantes do ambiente Unix, além do SIGSEGV, SIGUSR1, SIGUSR2, SIGALRM e SIGINT. Quais são suas características e utilidades?

SIGHUP (1 - hangup - corte): sinal emitido para processos filhos quando este se desconecta.

SIGQUIT (3 - quit - abandono): sinal emitido aos processos com tecla de abandono acionada (a.k.a CTRL+D).

SIGILL (4 - Instrução Ilegal): emitido em instrução ilegal.

SIGKILL (9): Destruição! FIM!

SIGTERM (10): Terminação por software.

3) Considere o código a seguir:

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

```
void tratamento_alarme(int sig)  
{  
    system("date");
```

```
    alarm(1);  
}  
  
int main()  
{  
    signal(SIGALRM, tratamento_alarme);  
    alarm(1);  
    printf("Aperte CTRL+C para acabar:\n");  
    while(1);  
    return 0;  
}
```

Sabendo que a função alarm() tem como entrada a quantidade de segundos para terminar a contagem, quão precisos são os alarmes criados neste código? De onde vem a imprecisão? Este é um método confiável para desenvolver aplicações em tempo real?

A contagem é relativamente precisa mas não é adequada para aplicações em tempo real já que apresentam alguma imprecisão devido a necessidade de executar outros processos no sistema operacional.