
Tratamento de sinais em sistemas UNIX

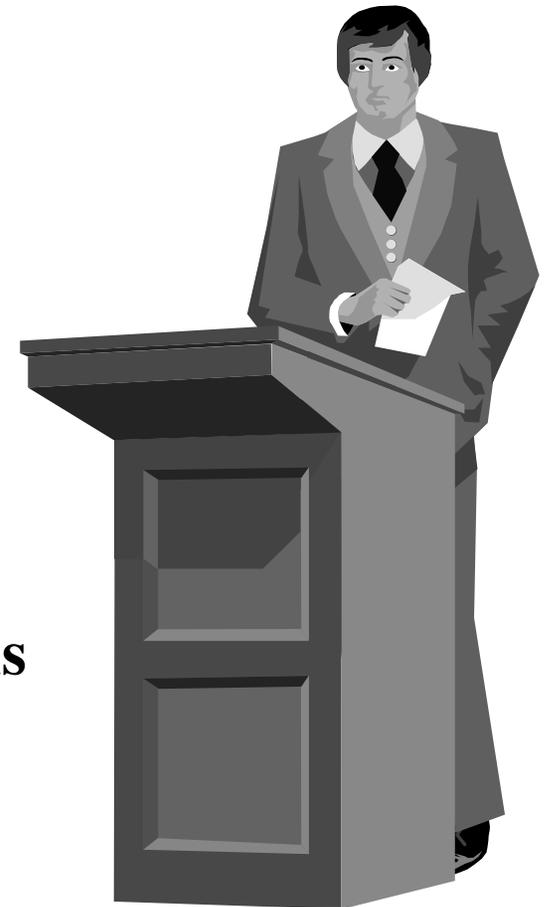
Volnys Borges Bernal

`volnys@lsi.usp.br`

`http://www.lsi.usp.br/~volnys`

Laboratório de Sistemas Integráveis

`http://www.lsi.usp.br/`



Agenda

- ❑ **Sinais no sistema UNIX**
- ❑ **Sinais gerados pelo teclado**
- ❑ **Utilitário Kill**

Sinais no sistema UNIX



Sinais no sistema UNIX

- **No sistema UNIX:**
 - **Tradução: sinal = signal**
 - **Sinais são eventos assíncronos que chegam aos processos**
 - **Os sinais podem ocorrer a qualquer momento**
 - **Existem vários tipos de sinais, cada um é representado por um número**
 - **A ação padrão decorrente da chegada de um processo pode ser (cada sinal possui uma ação default específica):**
 - **Term** Terminar o processo
 - **Ign** Ignorar o sinal
 - **Core** Terminar o processo e realizar o dump
 - **Stop** Suspender momentaneamente o processo
 - **Cont** Retomar a execução do processo
 - **O processo pode especificar uma ação a ser tomada no momento de chegada de um determinado sinal:**
 - Terminar o processo
 - Ignorar o sinal
 - Executar uma rotina (denominada rotina de tratamento do sinal)

Sinais no sistema UNIX

□ Sinais mais importantes:

1	SIGHUP	hung up - desconexão do terminal
2	SIGINT	interrupção vinda do teclado
3	SIGQUIT	quit vindo do teclado
	
9	SIGKILL	kill
	...	
14	SIGALRM	signal do temporizador, chamada alarm(2)
15	SIGTERM	terminação por software
	...	

Sinais no sistema UNIX

- ❑ **Sinais que podem ser enviados pelo terminal:**
(verifique a tecla em seu sistema executando `"stty -a"`)
 - **Sinal 2 - SIGINT**
 - Ação padrão: termina o processo
 - Este sinal pode ser tratado ou ignorado pelo processo
 - Pode ser enviado pelo teclado (geralmente tecla ^C)
 - **Sinal 3 - SIGQUIT**
 - Ação padrão: termina o processo com geração de "core"
 - Este sinal pode ser tratado ou ignorado pelo processo
 - Pode ser enviado pelo teclado (geralmente tecla ^\)

Sinais no sistema UNIX

- ❑ **Sinais que NÃO podem ser enviados pelo terminal:**
 - **Sinal 9 - SIGKILL**
 - Ação padrão: termina o processo
 - Este sinal NÃO pode ser tratado ou ignorado pelo processo

 - **Sinal 14 - SIGALRM**
 - Ação padrão: termina o processo
 - Este sinal pode ser tratado pelo processo

 - **Sinal 15 - SIGTERM**
 - Ação padrão: termina o processo
 - Este sinal pode ser tratado e ignorado pelo processo

Sinais gerados pelo teclado



Sinais gerados pelo teclado

- ❑ O comando `stty` mostra a sequencia de teclas para alguns sinais que podem ser enviados pelo teclado:

```
{terra|jose} stty -a
```

```
.....
```

```
.....
```

```
.....
```

```
{terra|jose}
```

Sinais gerados pelo teclado

□ Exemplo

- Encerrar um processo com sinal SIGINT via teclado

```
{terra|jose} ./loop
```

```
^C
```

```
{terra|jose}
```

Sinais gerados pelo teclado

□ Programa loop.c

```
#include <stdio.h>

int main()
{
    printf("Programa loop ... \n");
    while (1);
}
```

Sinais gerados pelo teclado

□ Exemplo

- Existem processos que podem “tratar” o sinal 2 (SIGINT)
- Neste caso, pode-se tentar enviar o sinal 3 (SIGQUIT)

```
{terra|jose} ./trata-intr
^C          <- enviando o sinal 2 (SIGINT)
^C          processo trata o sinal 2
^C
^\  
Quit (core dumped)
{terra|jose}
```

Sinais gerados pelo teclado

□ Programa trata_intr.c

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

void handler(int sig)
{
    printf("Chegou um sinal %d \n", sig);
}

int main()
{
    signal(2, handler);
    printf("Programa que trata o sinal INTR \n");
    printf("Aguardando ... \n");
    while (1);
}
```

Sinais gerados pelo teclado

□ Exemplo

- Existem processos que podem “tratar” o sinal 2 (SIGINT) e também o sinal 3 (SIGQUIT)

```
{terra|jose} ./trata-todos
^C          <- enviando o sinal 2 (SIGINT)
             processo trata o sinal 2
^C
^C
^\  
          <- enviando o sinal 3 (SIGQUIT)
^\  
          processo trata o sinal 3
....
```

□ Como terminar este processo?

Sinais gerados pelo teclado

□ Programa trata_todos.c

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

void handler(int sig)
{
    printf("Sinal %d \n", sig);
}

int main()
{
    int i;
    for (i=1; i<20; i++)
        signal(i, handler);
    printf("Programa trata todos os sinais\n");
    while (1);
}
```

Sinais gerados pelo teclado

- ❑ **Não é possível terminar este processo utilizando os sinais que podem ser encaminhados pelo teclado.**
- ❑ **Necessário encaminhar os sinais através do utilitário “kill”**

Utilitário Kill



Utilitário kill

❑ Descrição

- Utilitário que permite enviar sinais a outros processos
- Se não for especificado o número do sinal é enviado o sinal 15 (SIGTERM - terminação por software)

❑ Sintaxe

```
kill [-sinal] pid [...]
```

signal: nº do sinal

pid: process identification (nº do processo)

❑ Exemplo

```
# kill -INTR 813 812
```

```
# kill -2 813 812
```

Utilitário kill

❑ Dispare o programa trata_intr

```
{terra|jose} ./trata_intr  
Sinal 2
```

```
{terra|jose}
```

❑ Em outra janela execute:

```
{terra|jose} ps xau  
.....  
{terra|jose} kill -2 PID  
{terra|jose} kill -15 PID
```

Utilitário kill

❑ Dispare o programa trata_todos

```
{terra|jose} ./trata_todos  
Sinal 2  
Sinal 3  
Sinal 15  
{terra|jose}
```

❑ Em outra janela execute:

```
{terra|jose} ps xau  
.....  
{terra|jose} kill -2 PID  
{terra|jose} kill -3 PID  
{terra|jose} kill -15 PID  
{terra|jose} kill -9 PID
```