

INE5412 Sistemas Operacionais I

L. F. Friedrich

Tratamento de Sinais

O que é um Sinal?

Sinal é como uma **interrupção de software**

- É uma **entidade do kernel**.
- Ex, quando você pressiona Ctrl + C, o kernel envia o sinal SIGINT para o processo.
- Ctrl + C não é um um sinal, mas SIGINT é um sinal gerado quando Ctrl + C é pressionado.
- É puramente uma entidade de software.
 - Quando um evento relativo ao hardware acontece, um “sinal” é gerado do hardware para o kernel. Este “sinal” é chamado **interrupção de hardware**.
 - De outra forma, os sinais são chamados **interrupção de software**.

Dois tipos de Sinais

- Iniciados por **outros processos**.
 - SIGINT: Ctrl + C – Interrupção do teclado;
 - SIGTSTP: Ctrl + Z – sinal Stop do teclado;
 - SIGQUIT: Ctrl + \ - sinal Quit do teclado;
 - Usando programa “kill”;
 - Usando chamada de sistema kill().
- Iniciado pelo **kernel**.
 - SIGSEGV: sinal Segmentation fault;
 - SIGFPE: sinal de Floating point exception;
 - SIGCHLD: sinal de Child termination; etc.
- Um sinal dispara a execução de um **tratador de sinal**.

-
- Lista de sinais

Obs: man 5 signal e <sys/signal.h>

Obs: Alguns sinais não podem ser tratados nem ignorados (*)

Nome	Nº	Resposta padrão	Evento
SIGALARM	14	exit	relógio
SIGFPE	8	core & exit	exceção aritmética
SIGINT	2	exit	interrupção
SIGKILL (*)	9	exit	término
SIGPIPE	13	exit	rompimento de <i>pipe</i>
SIGQUIT	3	core & exit	término
SIGSEGV	11	core & exit	endereçamento inválido

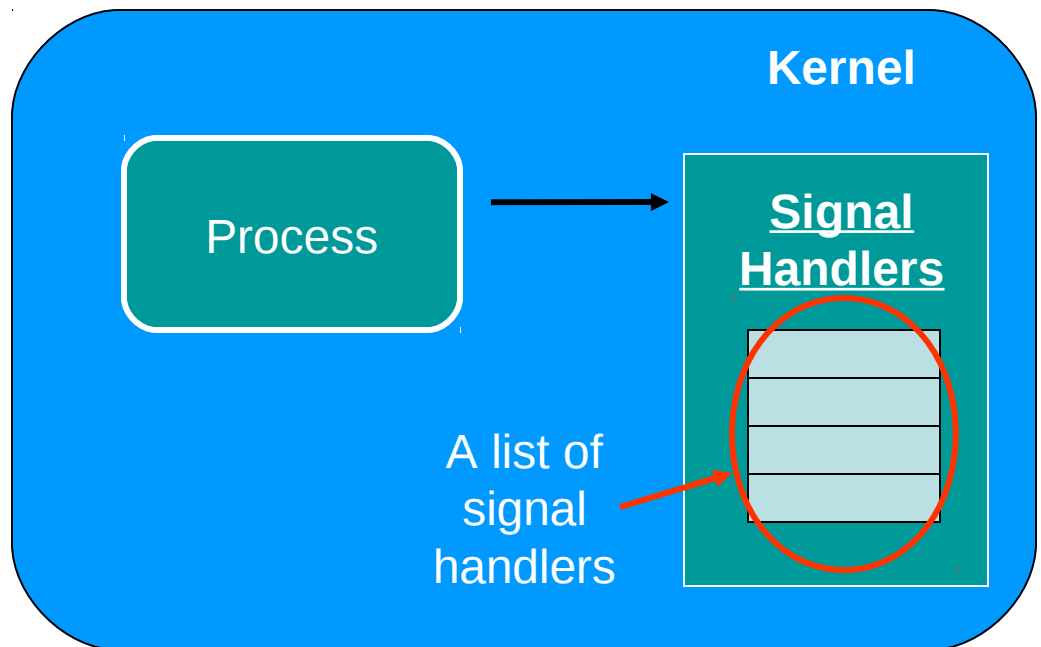
<http://www.cs.pitt.edu/~alanjawi/cs449/code/shell/UnixSignals.htm>

Tratador de Sinal

- Cada processo tem uma lista de tratadores de sinal correspondente a todos sinais disponíveis. Ex.,
 - SIGINT é associado a terminação; SIGTSTP é associado a suspensão.
 - Ver “man 7 signal” .

```
int main(void)
{
    while(1)
    {
        puts("hello");
    }
    puts("world");
}
```

Program Code

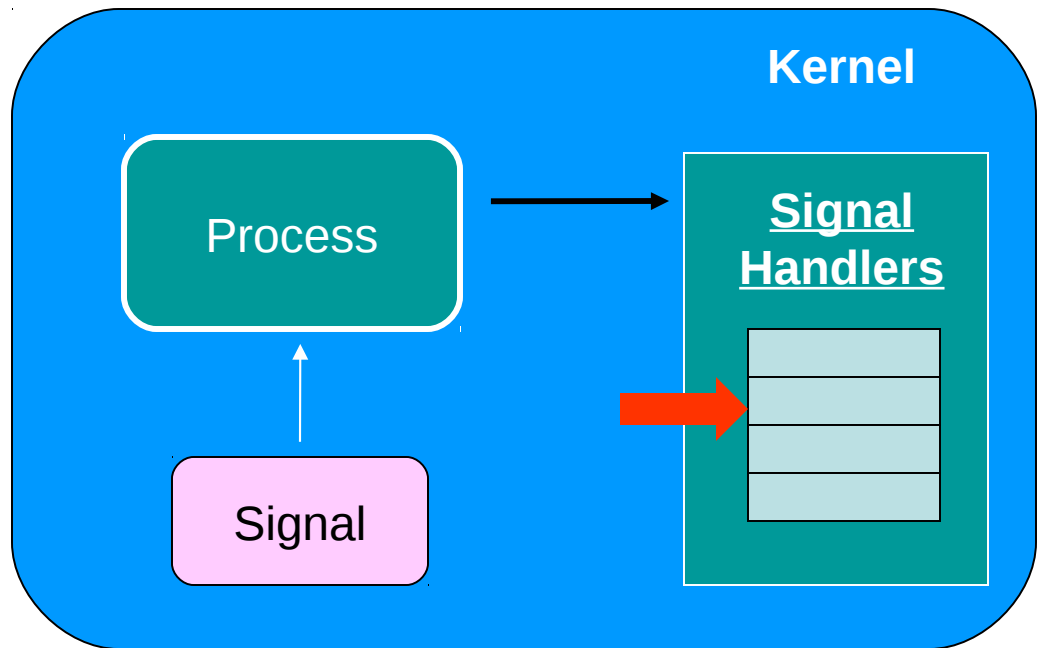


Tratador de Sinal

- Quando um sinal é **entregue** para o processo destino, a execução corrente do processo será parada.
- O kernel irá **mudar a execução** do processo para o correspondente tratador do sinal.

```
int main(void)
{
    while(1)
    {
        puts("hello");
    }
    puts("world");
}
```

Program Code

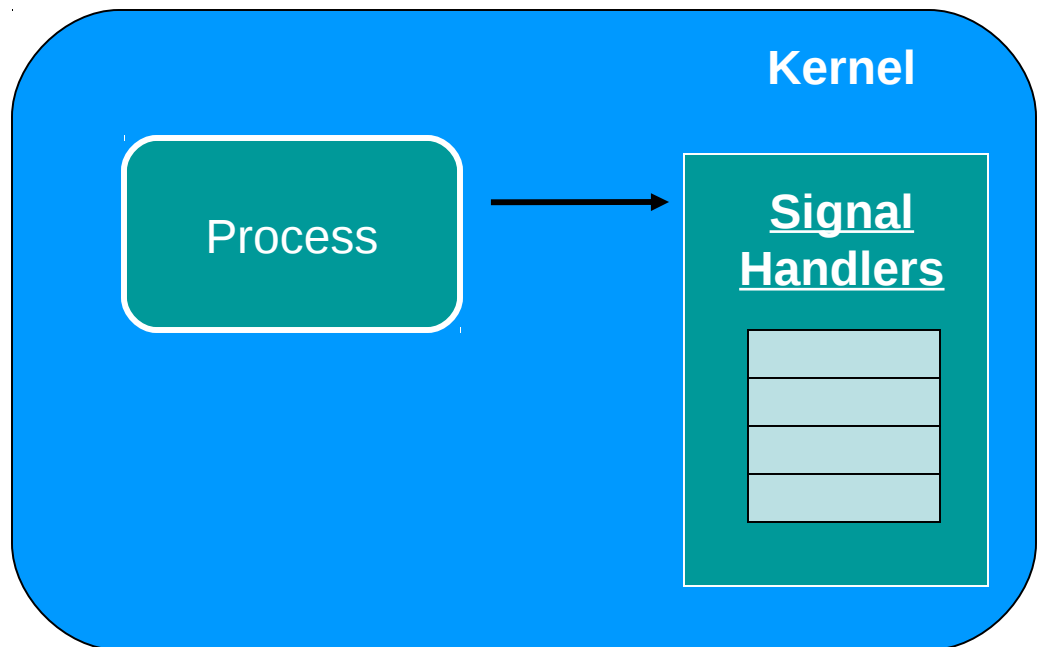


Tratador de Sinal

- Quando o tratador do sinal termina, o kernel **muda a execução** do processo de volta para o código do programa original.
 - Se o tratamento do sinal é para terminar o processo, então a execução do processo não pode mudar.

```
int main(void)
{
    while(1)
    {
        puts("hello");
    }
    puts("world");
}
```

Program Code



Como instalar tratador de sinal

- A chamada de sistema **signal()** instala um novo tratador de sinal para um sinal específico. O tratador deve ser uma função com o prototipo:

`void function_name(int signal_number)`

```
void sig_handler(int sig) {  
    if(sig == SIGINT)  
        puts("I received Ctrl + C");  
}
```

```
int main(void) {  
    signal(SIGINT, sig_handler);  
    printf("Press enter ");  
    getchar();  
    puts("End of program");  
}
```

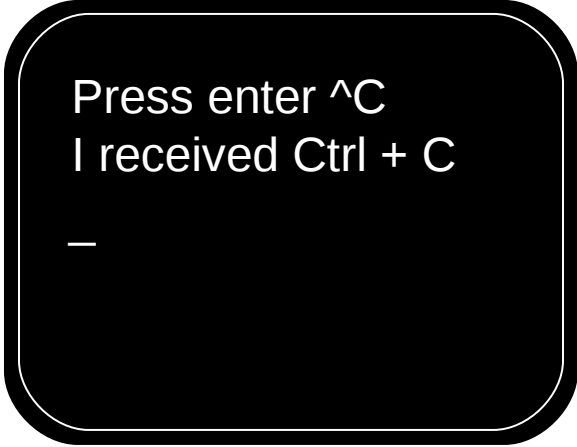


Tratador a ser instalado

Como instalar tratador de sinal

- Quando Ctrl + C pressionado, o tratador de sinal instalado será invocado.
 - O parametro passado para o tratador de sinal é o numero do sinal, que é SIGINT neste exemplo.

```
void sig_handler(int sig) {  
    if(sig == SIGINT)  
        puts("I received Ctrl + C");  
}  
  
int main(void) {  
    signal(SIGINT, sig_handler);  
    printf("Press enter ");  
    getchar();  
    puts("End of program");  
}
```




```
Press enter ^C  
I received Ctrl + C  
_
```

Como instalar tratador de sinal

- Finalmente, o processo retorna para onde ele recebeu o sinal.

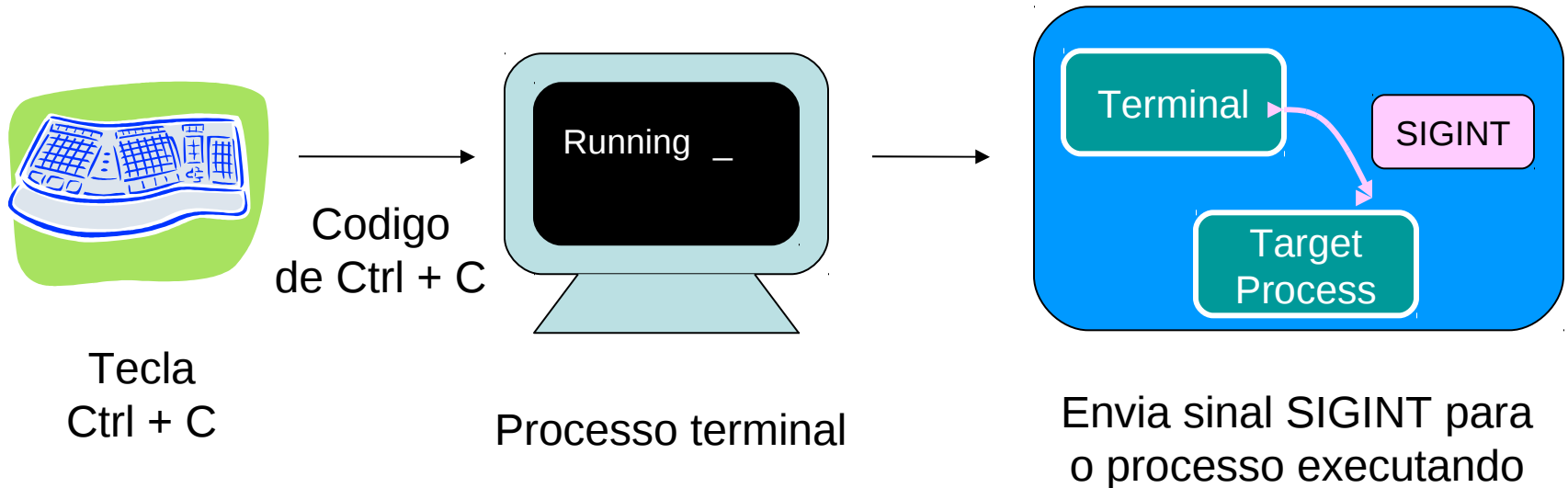
```
void sig_handler(int sig) {  
    if(sig == SIGINT)  
        puts("I received Ctrl + C");  
}  
  
int main(void) {  
    signal(SIGINT, sig_handler);  
    printf("Press enter ");  
    getchar();  
    puts("End of program");  
}
```



Press enter ^C
I received Ctrl + C
—

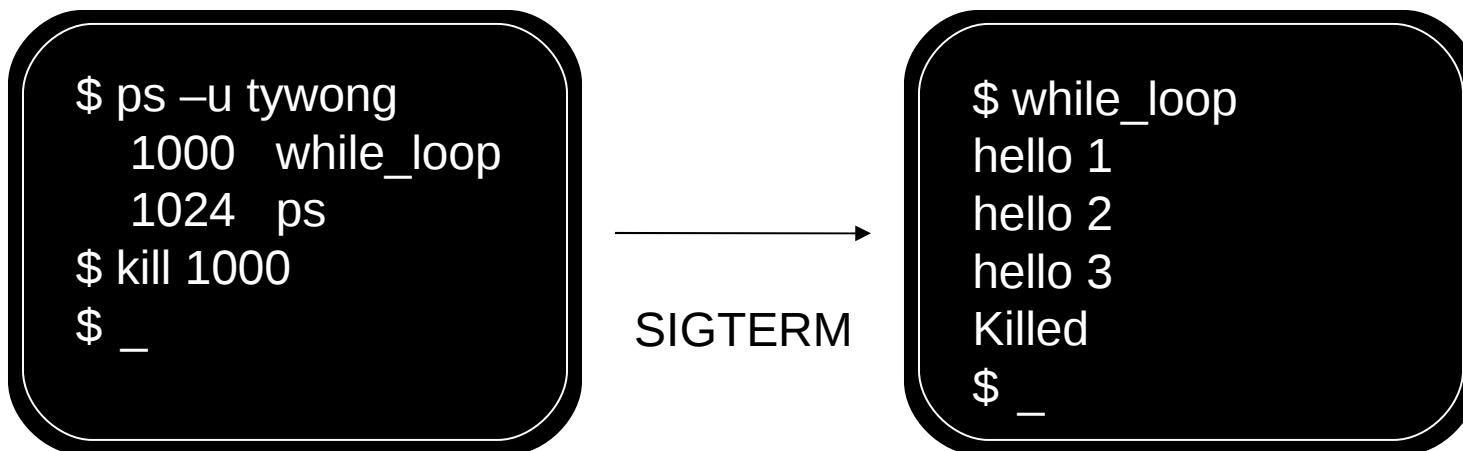
Como enviar sinal – Método 1

- Via terminal.
 - Pode enviar três sinais
 - SIGINT, SIGTSTP e SIGQUIT.



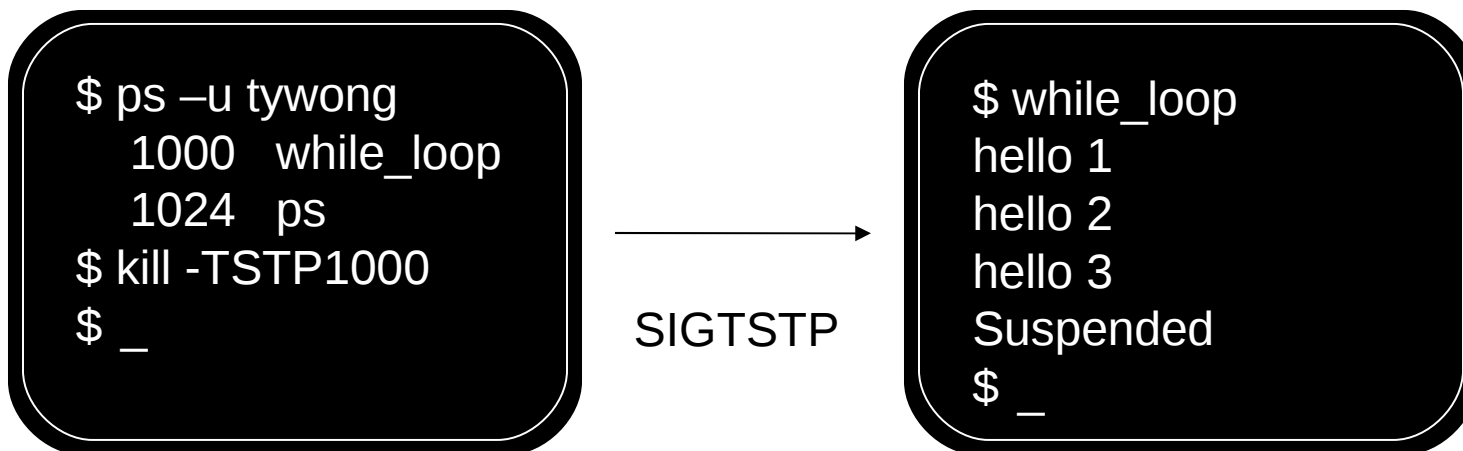
Como enviar sinal – Método 2

- Via comando “kill”.
 - Por default, o comando “kill” envia o sinal **SIGTERM** para o processo destino.
 - O tratamento default do SIGTERM é terminar o processo que recebe o sinal.



Como enviar sinal – Método 2

- O comando “kill” pode enviar qualquer sinal.
 - “kill -KILL [PID]”, ou “kill -9 [PID]”, envia o sinal SIGKILL para o processo destino e termina o mesmo **imediatamente**.
 - Outro exemplo pode ser SIGTSTP.



SIGKILL: <http://en.wikipedia.org/wiki/SIGKILL>

Como enviar sinal – Método 3

- Via a chamada de sistema `kill()`.
 - Esta chamada não é projetada para matar processos, mas para **enviar sinais** para processos.
 - Seu uso é similar ao do comando “kill”.
 - Ver em
 - `man 2 kill`.