

X36UNIX

UNIX - signály

jejich význam a použití

Martin Plicka - plickm1@fel.cvut.cz

Co jsou signály ?

- Softwarová přerušení zaslaná procesu jiným procesem, terminal driverem, jádrem
- Druhy signálů jsou očíslovány (od 0 výše) a pojmenovány. Seznam – např. **kill -1**
- **Synchronní** – původcem je ten samý proces (nepovolený zápis do paměti, nepovolená instrukce, dělení nulou ...)
- **Asynchronní** – původcem je jiný proces, terminálový driver (uživatel z klávesnice, ukončení spojení, shutdown ...)

Reakce na signály

Výchozí:

- Terminate (Exit) - přerušení procesu
- Core - přerušení a vytvoření souboru **core** s výpisem použité paměti
- Ignore - ignorování signálu
- Stop - zastavení procesu

Vlastní:

- Odchycení a vlastní obsluha (ignorování, vlastní rutina...), viz např. **man signal(3C)**, **sigaction(2)**
- Signály **SIGKILL**, **SIGSTOP** nelze odchytit, ignorovat

Příklady signálů – man signal(3HEAD)

Name	Value	Default	Event
SIGHUP	1	Exit	Hangup (see termio(7I))
SIGINT	2	Exit	Interrupt (see termio(7I))
SIGQUIT	3	Core	Quit (see termio(7I))
SIGILL	4	Core	Illegal Instruction
SIGTRAP	5	Core	Trace or Breakpoint Trap
SIGABRT	6	Core	Abort
SIGEMT	7	Core	Emulation Trap
SIGFPE	8	Core	Arithmetic Exception
SIGKILL	9	Exit	Killed
SIGBUS	10	Core	Bus Error
SIGSEGV	11	Core	Segmentation Fault
SIGSYS	12	Core	Bad System Call
SIGPIPE	13	Exit	Broken Pipe
SIGALRM	14	Exit	Alarm Clock
SIGTERM	15	Exit	Terminated
SIGUSR1	16	Exit	User Signal 1

... atd ... (viz **man -s 3HEAD signal**)

+ signál SIGEXIT (0) - je odeslán procesem sám sobě při ukončení

Příklad - synchronní signály

SIGFPE(8):

např. dělení nulou

```
[plickm1@sunray1:~]$ ./fpe
Arithmetic Exception
[plickm1@sunray1:~]$
```

fpe.c:

```
int a;
int main () {
    a = 5/0; // !!!
}
```

SIGSEGV(11):

nepovolený zápis do paměti

```
[plickm1@sunray1:~]$ ./segv
Segmentation Fault
[plickm1@sunray1:~]$
```

segv.c:

```
int * a;
int main () {
    a = (int *) 0;
    a [1] = 5; // !!!
}
```

15 - SIGTERM

- Standardní signál pro předčasné přerušeni procesu
- Proces jej může odchytit a vykonat vlastní rutinu, ignorovat
- Použit při volání příkazů `kill`, `killall` apod. bez explicitního výběru signálu

9 - SIGKILL

- Standardní signál pro předčasné násilné přerušeni procesu.
- Proces nemá šanci jej odchytit, ignorovat

SIGTERM & SIGKILL

- Používají se i např. na konci skriptů pro vypnutí systému, zde soubor Linuxu: **/etc/rc0.d/S90halt**

```
...  
  
# Now kill them all  
killall5 -15  
sleep 1  
echo -n "${BLUE}Sent all processes the TERM signal...${NORMAL}"  
mysleep "$BLUE"  
  
killall5 -9  
sleep 1  
echo -n "${RED}Sent all processes the KILL signal...${NORMAL}"  
mysleep "$RED"  
  
...
```

killall5 - zasílá v **Linuxu** signály všem běžícím procesům vyjma sebe a rodiče (**killall** v Linuxu a Solarisu se liší !!!)

1 – SIGHUP

Historie:

- Připojení terminálu bylo realizováno pomocí modemu nebo seriové linky. Při rozpojení (HangUP = „zavěšení“) bylo nutno informovat programy

Dnes:

- Při zavření terminálu se odešle SIGHUP běžícímu shellu, ten jej zachytává a ukončuje své potomky
- Proces jej při svém ukončení rozesílá potomkům
- Alternativně lze u démonů (např. Apache) signálem SIGHUP vynutit znovunačtení konfigurace

Příkaz NOHUP

- Nucené ukončení procesu při ukončení shellu (odhlášení) lze obejít použitím příkazu **nohup**.

nohup nas_prikaz &

- Proces poběží i po zavření shellu
- Případný výstup programu je směrován do souboru **nohup.out**

2 – SIGINT

- Je posílán terminálovým driverem po stisku nadefinovaných kláves (stand. **<CTRL+C>**)
- Standardní akce je ukončení

3 – SIGQUIT

- Je posílán terminálovým driverem po stisku nadefinovaných kláves (stand. **<CTRL+\\>**)
- Standardní akce je ukončení a výpis paměti do souboru (**Core**)

Příkaz STTY

- Slouží ke konfiguraci terminálu, mj. k určení kláves pro signály SIGINT a SIGQUIT

```
[plickm1@sunray1:test]# nastaví CTRL+B pro SIGINT  
[plickm1@sunray1:test]stty intr ^B
```

```
[plickm1@sunray1:test]# vypíše aktuální nastavení  
[plickm1@sunray1:test]stty -a
```

```
speed 9600 baud;  
rows = 24; columns = 80; ypixels = 0; xpixels = 0;  
csdata ?  
eucw 1:0:0:0, scrw 1:0:0:0  
intr = ^b; quit = ^\; erase = ^?; kill = ^u;  
eol = ^d; eol2 = <undef>; eol12 = <undef>; swtch = <undef>;  
start = ^q; stop = ^s; susp = ^z; dsusp = ^y;  
rprnt = ^r; flush = ^o; werase = ^w; lnext = ^v;  
...atd...
```

Příkaz KILL

- Odešle signál procesu nebo úloze

Syntaxe (Bash built-in):

```
kill [-s sigspec | -n signum | -sigspec] [pid | job] ...  
nebo  
kill -l [sigspec | exit_status]
```

Příklady:

<code>kill 123</code>	- odešle SIGTERM(15) procesu s PID=123
<code>kill %1</code>	- odešle SIGTERM(15) úloze s ID=1
<code>kill -KILL 123</code>	- odešle SIGKILL procesu s PID=123
<code>kill -n 9 123 456</code>	- odešle SIGKILL procesům s PID=123 a 456
<code>kill -l</code>	- vypíše tabulku existujících signálů
<code>kill -l HUP</code>	- vypíše číslo signálu SIGHUP
<code>kill -l 3</code>	- vypíše jméno signálu s č. 3

Pozn.: Existuje více implementací příkazu kill.

Příkaz TRAP

- Nastaví uživatelskou odezvu na signál v shellu

Syntaxe (Bash built-in):

```
trap [-lp] [arg] [sigspec ...]
```

Nastaví spuštění příkazu **arg** při obdržení zadaného signálu.

Příklady:

<code>trap -l</code>	- vypíše seznam existujících signálů
<code>trap</code>	- vypíše seznam nastavených odezev
<code>trap "echo Int" INT</code>	- nastaví odezvu pro SIGINT (vypíše „Int“)
<code>trap "" INT</code>	- nastaví ignorování SIGINT
<code>trap - INT</code>	- vrátí výchozí odezvu pro SIGINT
<code>trap -p INT SEGV</code>	- vypíše seznam nastavení pro dané signály

ZDROJE ?

Manuálové stránky:

- `signal (3HEAD)`
 - `bash`
- ... a další

WWW:

<http://www.wikipedia.org>

hesla „SIGHUP“ a jiné signály