Trap statement - Linux Bash Shell Scripting Tutorial Wiki

Archived: 2026-04-05 12:53:25 UTC

← Shell signal values • **Home** • How to clear trap →

- While running a script user may press Break or CTRL+C to terminate the process.
- User can also stop the process by pressing CTRL+Z.
- Error can occur do to bug in a shell script such as arithmetic overflow.
- This may result into errors or unpredictable output.
- Whenever user interrupts a signal is send to the command or the script.
- Signals force the script to exit.
- However, the trap command captures an interrupt.
- The trap command provides the script to captures an interrupt (signal) and then clean it up within the script.

Syntax

The syntax is as follows

```
trap arg signal
trap command signal
trap 'action' signal1 signal2 signalN
trap 'action' SIGINT
trap 'action' SIGTERM SIGINT SIGFPE SIGSTP
trap 'action' 15 2 8 20
```

Example

Create a shell script called testtrap.sh:

```
#!/bin/bash
# capture an interrupt # 0
trap 'echo "Exit 0 signal detected..."' 0

# display something
echo "This is a test"

# exit shell script with 0 signal
exit 0
```

Save and close the file. Run it as follows:

```
chmod +x testtrap.sh
./testtrap.sh
```

Sample outputs:

```
This is a test
Exit 0 signal detected...
```

- The first line sets a trap when script tries to exit with status 0.
- Then script exits the shell with 0, which would result in running echo command.
- Try the following example at a shell prompt (make sure /tmp/rap54ibs2sap.txt doesn't exits).
- Define a shell variable called $file:

```
file=/tmp/rap54ibs2sap.txt
```

Now, try to remove $file, enter:

Sample output:

```
rm: cannot remove `/tmp/rap54ibs2sap.txt': No such file or directory
```

Now sets a trap for rm command:

```
trap "rm $file; exit" 0 1 2 3 15
```

Display list of defined traps, enter:

Sample outputs:

```
trap -- 'rm /tmp/rap54ibs2sap.txt; exit' EXIT
trap -- 'rm /tmp/rap54ibs2sap.txt; exit' SIGHUP
trap -- 'rm /tmp/rap54ibs2sap.txt; exit' SIGINT
trap -- 'rm /tmp/rap54ibs2sap.txt; exit' SIGQUIT
trap -- 'rm /tmp/rap54ibs2sap.txt; exit' SIGTERM
```

Now, try again to remove the $file, enter:

This time rm command did not displayed an error. The $file doesn't exist yet. The trap command simply exit whenever it get 0, 1, 2, 3, or 15 signal. Try capturing CTRL+C:

```
#!/bin/bash
# capture an interrupt # 2 (SIGINT)
```

```
trap '' 2
# read CTRL+C from keyboard with 30 second timeout
read -t 30 -p  "I'm sleeping hit CTRL+C to exit..."
```

Sample outputs:

```
I'm sleeping hit CTRL+C to exit...^C^C^C^C
```

← Shell signal values • **Home** • How to clear trap →
