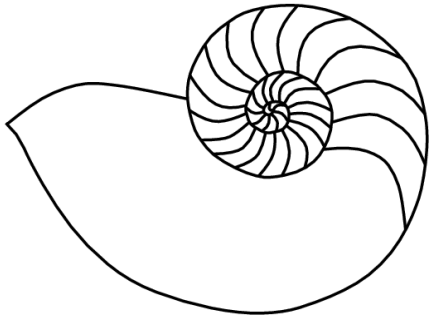
University of Manitoba | Computer Science

UNIX Command
RTFM: trap(1p)

Gilbert Detillieux

October 8, 2013
MUUG Meeting




THE
BOURNE
SHELL TRAP

What is “trap”?...

- Built-in command in Bourne Shell and derivatives (ksh, bash, etc.)
- Used for signal(2) handling in scripts
- Can also be used to trap certain actions (e.g. EXIT and ERR)

trap(1p) Man Page

- NAME
  - trap - trap signals
- SYNOPSIS
  - trap [action condition ...]
- DESCRIPTION
  - If action is “-”, the shell shall reset each condition to the default value. If action is null (""), the shell shall ignore each specified condition if it arises. Otherwise, the argument action shall be read and executed by the shell when one of the corresponding conditions arises.
  - The condition can be EXIT, 0 (equivalent to EXIT), or a signal specified using a symbolic name, without the SIG prefix, as listed in the tables of signal names in the <signal.h> header.

Trap Actions

- Null string (“”) means ignore the signal/condition
- Hyphen (“-”) means revert to previous action
- Anything else is a command to evaluate

Trap Conditions

- 0 (or EXIT) triggered on shell process exit
- Anything else is a <signal.h> number
- Can specify numerically
  - E.g. 1 = SIGHUP, 2 = SIGINT, etc.
  - Backward-compatible to original Bourne shell
  - But less portable to different systems
- Can be symbolic name, minus “SIG”
  - E.g. SIGTERM -> TERM, SIGHUP -> HUP, etc.
- Some implementations allow
  - “SIG” prefix (SIGINT or INT)
  - Lower case (SIGHUP or sighup or hup)
  - May not be POSIX compliant or as portable

Trap Examples

- trap ‘rm –f “$TMP”; exit’ EXIT
  - On exit, remove temporary files
- trap exit INT HUP TERM
  - On receipt of listed signals, run exit command
- trap – INT HUP TERM
  - Revert to default action on listed signals
- trap INT
  - Same, but limited to one argument

Ignoring Signals…

- trap ‘’ INT
  - Completely ignore signal
  - Inherited by child processes
- trap : INT *or* trap true INT
  - Trap signal, but don’t do anything
  - Child processes revert to default action

Watch your quoting!...

- Arguments evaluated twice:
  - When “trap” command is evaluated
  - When action is triggered
- trap “rm –f $TMP” EXIT
  - $TMP expanded when “trap” command run
- trap ‘rm –f $TMP’ EXIT
  - $TMP expanded when action triggered

Trap without arguments:

- bash-3.2$ trap
- bash-3.2$ trap 'echo $RANDOM' INT
- bash-3.2$ trap
- trap -- 'echo $RANDOM' SIGINT
- bash-3.2$ trap "echo $RANDOM" HUP
- bash-3.2$ trap
- trap -- 'echo 5432' SIGHUP
- trap -- 'echo $RANDOM' SIGINT
- bash-3.2$

Save & Restore State

- save_state = `trap`
- trap ‘do_stuff’ INT HUP TERM
- …
- eval “$save_state”

Bash Extensions

- trap –p
  - Print list of trap commands
  - Same as if no arguments given
- trap –p *condition* …
  - Print list of trap commands for given conditions (signals)
- trap –l
  - Print list of allowed signals
  - Highly system-dependent

Bash Extensions

- trap '*action*' ERR
  - executed whenever a command has non-zero exit status, except if command is part of…
    - command list immediately following **while** or **until** keyword,
    - part of the test in an **if** statement,
    - part of a **&&** or **??** list,
    - or if command's return value is being inverted via **!**.
- trap '*action*' RETURN
  - executed at end of each…
    - shell function or
    - script executed with the **.** or **source** builtins.
- trap '*action*' DEBUG
  - executed before every
    - simple command,
    - **for** command, **case** command, **select** command, arithmetic **for** command, and
    - first command executes in a shell function

Questions?