



Script Recording

(Scripting Terminal Sessions)

Presentation shared with the

St. Louis Unix Users Group

2023-11-08

Welcome



Introduction

- https://www.sluug.org/bio/Stan_Reichardt
- Benevolent dictator for life of Newlinux.org .
 - Arbitrary, cranky, profane, not politically correct, and old.
 - Acronym, name and term pronunciation use is by Missouri rules.
 - Tends to get terminology mixed up (Phil and Ed correct him) .
 - Very much a Linux dogmatist (near Linux bigot) .
 - Wears belt and suspenders.





Who uses script

- A commonly used scripting technique uses history of keyboard input.
 - The hidden `~/.history` file is frequently extracted and used.
 - `~$ man 3 history`
 - `~$ man, or info, or pinfo history (3)`
- Using **script** and affiliated commands are not well known.
- Only used by someone that knows about them.
- On Unix and Linux (don't know about other Operating Systems) .



What commands

Three related commands:

- **script** (1) - make typescript of terminal session
- **scriptlive** (1)- re-run session typescripts, using timing information
- **scriptreplay** (1) - play back typescripts, using timing information



What purpose

- Document what you do.
- Build executable shell scripts using **script** with what you already do.
 - Built by recording your keystrokes.
 - Visible history of the commands you give for execution.



Why use script?

- The **script** command will record the terminal.
 - Captures what was typed.
 - Captures results.
 - Can be played back in real time (accelerated or slowed) .
- Can document, prove or teach how a job/project/step/task was done.
- It is readily available (already installed on many distributions) .
- Can be used where there is no graphical environment (no GUI) .
 - Across a simple SSH connection.
 - On servers.



When to use

- Do NOT to use:
 - When you will be using a Graphical User Interface (GUI) tool.
 - The GUI will cause capture of too many problematic globs.
 - Should severely limit use to Command Line Interface (CLI) only.
 - When you want to use something more. Like SimpleScreenRecorder.
- When to use.
 - When you want more than just your history files.
 - When you want to capture results of commands that you ran.
 - On a server, where you should not load graphical tools.
 - Ad hoc, when needing something quickly (is usually already installed) .
 - Safely used within terminals run within GUI sessions.



Available

- The **script** command is part of the **util-linux** package which can be downloaded from Linux Kernel Archive
 - `<https://www.kernel.org/pub/linux/utils/util-linux/>`
- The **util-linux** package is not always automatically installed.
- My experience has been that it is usually available.



GUI Alternatives

- In distant past I used proprietary package on MS Windows XP.
 - So long ago, that I don't remember it's name (nor care) .
- My full screen GUI capture choice is **SimpleScreenRecorder**.
 - <https://www.maartenbaert.be/simplescreenrecorder/>
 - <https://github.com/MaartenBaert/ssr/>
 - Joe Collins, EzeeLinux.com recommended and uses this package.
 - Free, Linux only and used with X Windows System.
- Encroachment by Wayland may become a problem.
- To find alternatives:
 - <https://alternativeto.net/> (114 alternatives listed) .



How to get started

- How to get started (RTFM both man pages and info pages) .
 - Find the man page for script (NO EXAMPLES)
 - ~\$ man script
 - ~\$ info script
 - ~\$ **pinfo** script (my preferred way to view man pages and/or info pages) .
 - Looking for relevant man pages (~\$ apropos script -- gives 121 results) .
 - Just relevant man pages for script (use ~\$ apropos typescript) .
 - script (1) - make typescript of terminal session (NO EXAMPLES)
 - scriptlive (1) - re-run session typescripts, using timing information
 - scriptreplay (1) - play back typescripts, using timing information



EXAMPLE from `scriptreplay`

- EXAMPLE from `scriptreplay` man page :

```
% script --log-timing file.tm --log-out script.out
```

```
Script started, file is script.out
```

```
% ls
```

```
<etc, etc>
```

```
% exit
```

```
Script done, file is script.out
```

```
% scriptreplay --log-timing file.tm --log-out script.out
```



EXAMPLE from scriptlive

- EXAMPLE from **scriptlive** man page:

```
% script --log-timing file.tm --log-in script.in
```

```
Script started, file is script.out ## ( ? )
```

```
% date
```

```
<etc, etc>
```

```
% exit
```

```
Script done, file is script.out ## ( ? )
```

```
% scriptlive --log-timing file.tm --log-in script.in
```

- Did NOT look right to me. Not understanding results.



How to capture

- How to get started
 - Example:
- Build your own scripts
- How to save and exit
- How to abort if you screw up
- How to fix errors.
- How to write code with script



How to play back

- How to play back
 - Reconstruct
 - Trace
- How to set with timing information.



Demonstration examples

- This may be optional slide and might be placed earlier.
- If there is enough time left, we will try to demonstrate.
- Following slides may just be reminders of what you want to demo.



Script Recording

What are your questions?

Presentation shared with the

St. Louis Unix Users Group

2023-11-08

Questions