

# Shell Spells

Thamme Gowda, <http://isi.edu/~tg>

# Table of Contents

GNU Parallel .....	2
Clear Terminal .....	3
Invert / Exclude files .....	4
Bash Exit On Error .....	5
Get Free Port .....	6
Find Files Modified in Last Hour .....	7



# spell<sup>2</sup>

/spel/

*noun*

noun: **spell**; plural noun: **spells**

a form of words used as a magical charm or incantation.

"a spell is laid on the door to prevent entry"

ಸಮಾನ:

incantation

charm

conjunction

rune

magic formula



- a state of enchantment caused by a magic spell.  
"the magician may **cast a spell on** himself"
- an ability to control or influence people as though one had magical power over them.  
"she is afraid that you are waking from her spell"

ಸಮಾನ:

irresistible influence

fascination

magnetism

animal magnetism



*Spell, /spel/, noun, a form of words used as a magical charm or incantation.*

A bunch of unix spells, sorted in no definite order.

# GNU Parallel

Split input into parts having no more than 5 lines each, but run j=2 jobs in parallel, and keep input order

```
seq 100 | parallel --keep-order --pipe -N 5 -j 2 'wc -lw && sleep 2 '
```

Show progress

```
seq 100 | parallel --progress --keep-order --pipe -N 5 -j 2 'wc -lw && sleep 2 '
```

# Clear Terminal

We all use `clear` or `CTRL+L` but it's not really a "clear" terminal.

```
printf "\033c"    #<ESC>c  
reset            # another way
```

Info [stackoverflow.com/q/5367068/1506477](https://stackoverflow.com/q/5367068/1506477)

# Invert / Exclude files

Info : [www.tecmint.com/delete-all-files-in-directory-except-one-few-file-extensions/](http://www.tecmint.com/delete-all-files-in-directory-except-one-few-file-extensions/)

```
shopt -s extglob # enable
ls !(conf.yml)   # list all except conf.yml
rm -r !(conf.yml) # rm all except conf.yml
rm -v !("filename1"|"filename2") # multiple exclusions
```

# Bash Exit On Error

```
set -euo pipefail # recommended
set -euxo pipefail # Adding "x" prints out debugging statement
```

**-e** exits on error, **-u** errors on undefined variables, **-x** for echoing each statement, and **-o pipefail** exits on command pipe failures.

If **set -x** wasn't enough to debug, and you like to get more info

```
export PS4='+(${BASH_SOURCE}:${LINENO}): ${FUNCNAME[0]:+${FUNCNAME[0]}(): }'
```

1. [vaneyckt.io/posts/safer\\_bash\\_scripts\\_with\\_set\\_euxo\\_pipefail/](https://vaneyckt.io/posts/safer_bash_scripts_with_set_euxo_pipefail/)
2. [stackoverflow.com/a/2871034/1506477](https://stackoverflow.com/a/2871034/1506477)
3. [stackoverflow.com/a/17805088/1506477](https://stackoverflow.com/a/17805088/1506477)

# Get Free Port

Trying to bind to port `0` makes the kernel allocate a free port.

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind('', 0)
addr = s.getsockname()
print(addr[1])
s.close()
```

Shell one-liner:

```
PORT=$(python -c 'import socket; s=socket.socket(); s.bind("", 0);
print(s.getsockname()[1]); s.close()')
echo $PORT
```



# Find Files Modified in Last Hour

[stackoverflow.com/a/33407491/1506477](https://stackoverflow.com/a/33407491/1506477)

```
find . -mmin -60 -type f -exec ls -l {} +
```