Big Data Processing with Unix Tools

Diomidis Spinellis
Professor of Software Analytics
Department of Software Technology
Delft University of Technology

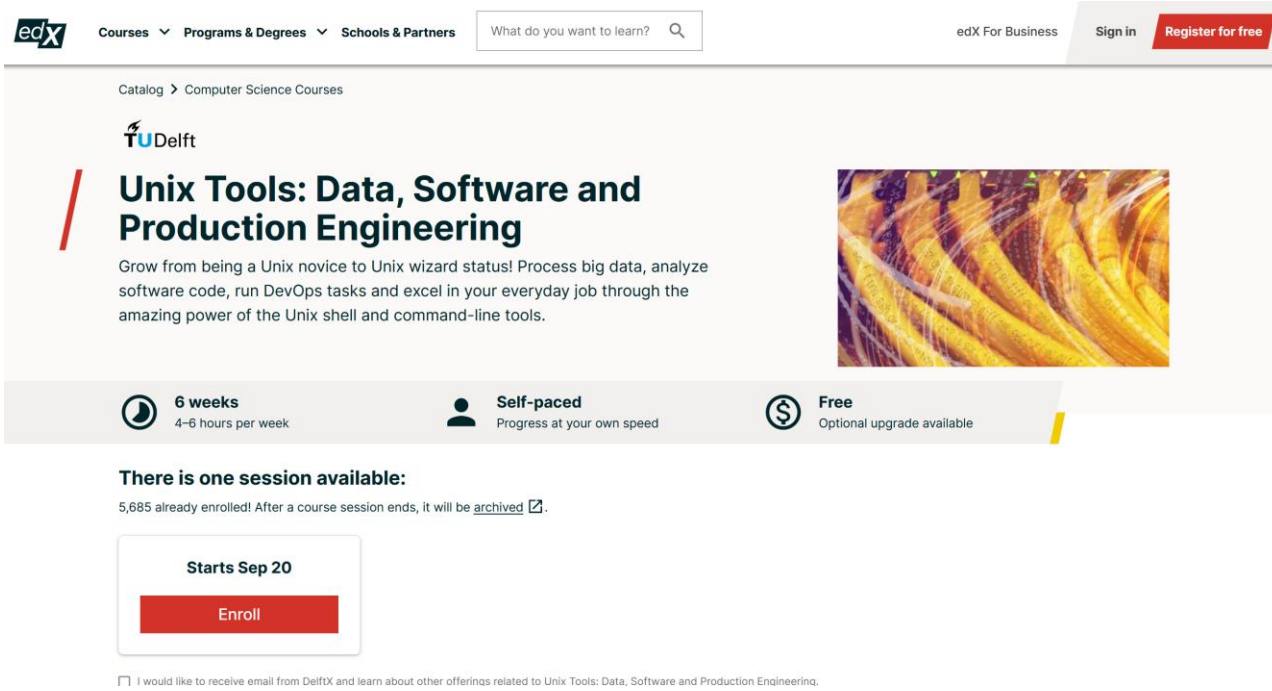
http://www.spinellis.gr/
D.Spinellis@tudelft.nl

Overview

Foundations

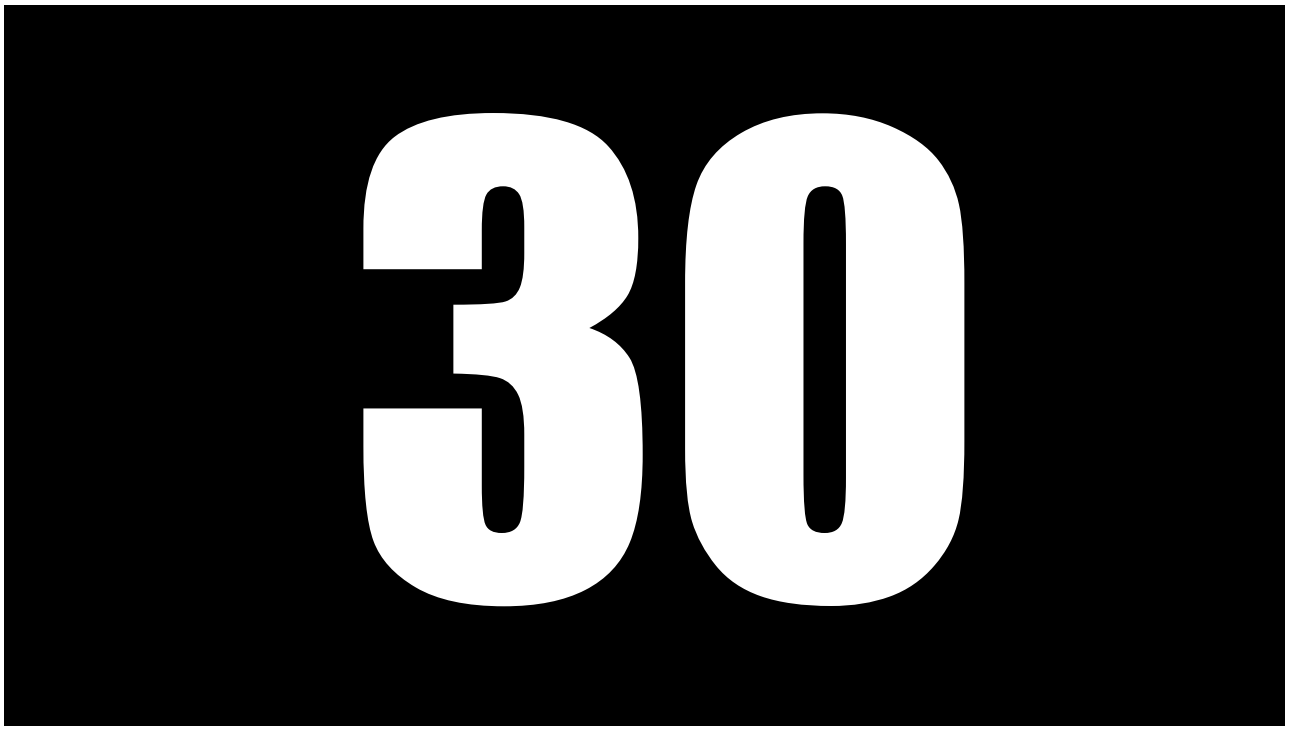
Data fetching and selection

Data processing and reporting


edX
Courses
Programs & Degrees
Schools & Partners
What do you want to learn?
edX For Business
Sign in
Register for free
Catalog > Computer Science Courses
TUDelft
Unix Tools: Data, Software and Production Engineering
Grow from being a Unix novice to Unix wizard status! Process big data, analyze software code, run DevOps tasks and excel in your everyday job through the amazing power of the Unix shell and command-line tools.
6 weeks
4–6 hours per week
Self-paced
Progress at your own speed
Free
Optional upgrade available
There is one session available:
5,685 already enrolled! After a course session ends, it will be archived.
Starts Sep 20
Enroll
I would like to receive email from DelftX and learn about other offerings related to Unix Tools: Data, Software and Production Engineering.

3

4



5

Advantages

6

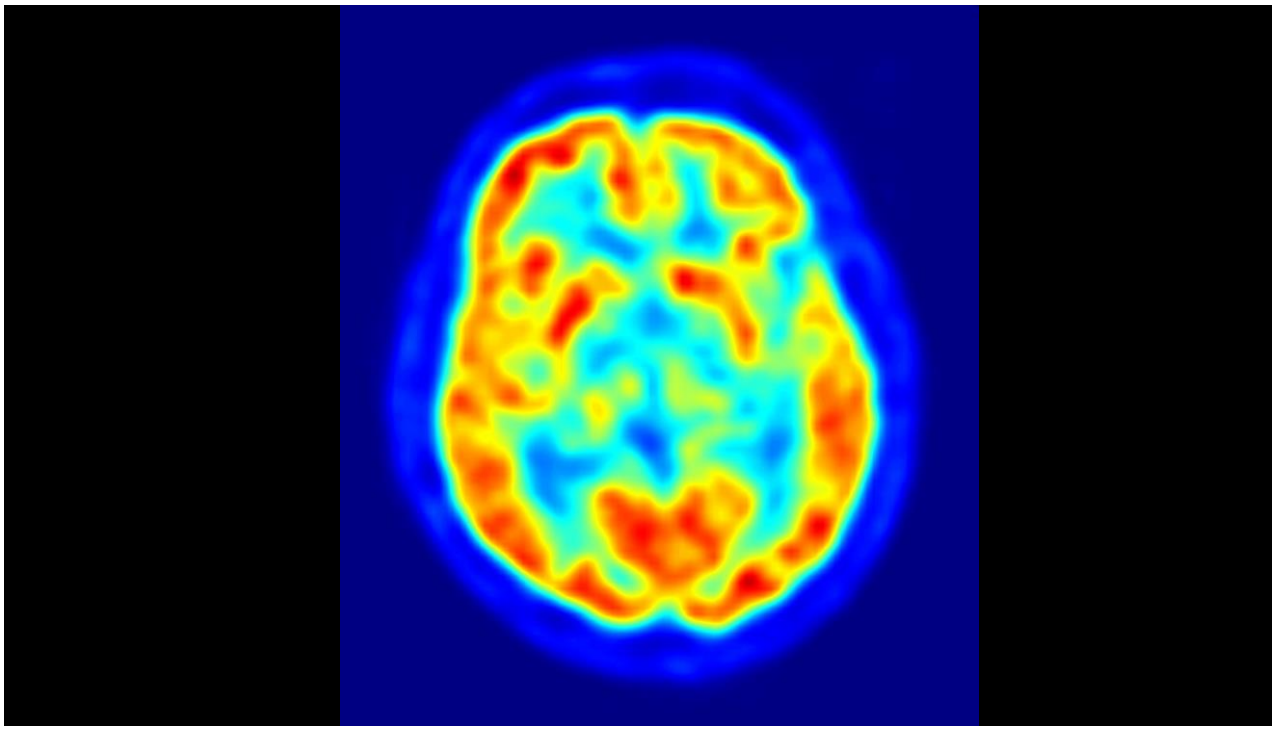
Efficient

7

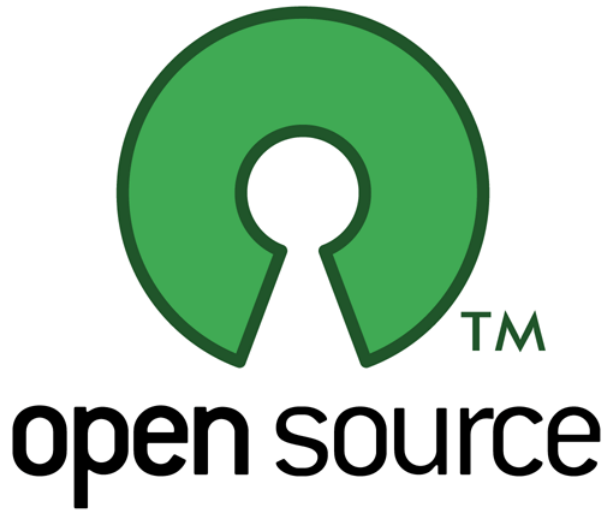
8

9

10



11



12

Use Cases

- Exploratory / interactive data analytics
- Large data sets
- Heterogeneous data

small is beautiful

15

```
main(int argc, char *argv[])
 {

     while (*argv) {
            (void)printf("%s", *argv);
            if (*++argv)
                  putchar(' ');
     }
    putchar('\n');
     exit(0);
}
```

16

```
# $FreeBSD: src/etc/master.passwd,v 1.40 2005/06/06 20:19:56 brooks Exp $
#
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5:System &:/:/usr/sbin/nologin
bin:*:3:7:Binaries Commands and Source:/:/usr/sbin/nologin
tty:*:4:65533:Tty Sandbox:/:/usr/sbin/nologin
mem:*:5:65533:KMem Sandbox:/:/usr/sbin/nologin
games:*:7:13:Games pseudo-user:/usr/games:/usr/sbin/nologin
news:*:8:8:News Subsystem:/:/usr/sbin/nologin
man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin
sshd:*:22:22:Secure Shell Daemon:/var/empty:/usr/sbin/nologin
smmsp:*:25:25:Sendmail Submission
User:/var/spool/clientmqueue:/usr/sbin/nologin
mailnull:*:26:26:Sendmail Default
User:/var/spool/mqueue:/usr/sbin/nologin
bind:*:53:53:Bind Sandbox:/:/usr/sbin/nologin
proxy:*:62:62:Packet Filter pseudo-user:/nonexistent:/usr/sbin/nologin
_pflogd:*:64:64:pflogd privsep user:/var/empty:/usr/sbin/nologin
_dhcp:*:65:65:dhcp programs:/var/empty:/usr/sbin/nologin
uucp:*:66:66:UUCP pseudo-
user:/var/spool/uucppublic:/usr/local/libexec/uucp/uucico
pop:*:68:6:Post Office Owner:/nonexistent:/usr/sbin/nologin
www:*:80:80:World Wide Web Owner:/nonexistent:/usr/sbin/nologin
```

17

18

Data Engineering

- Extract
- Transform
- Load

21

Extract

22

23

Select

24

25

Process

26

27

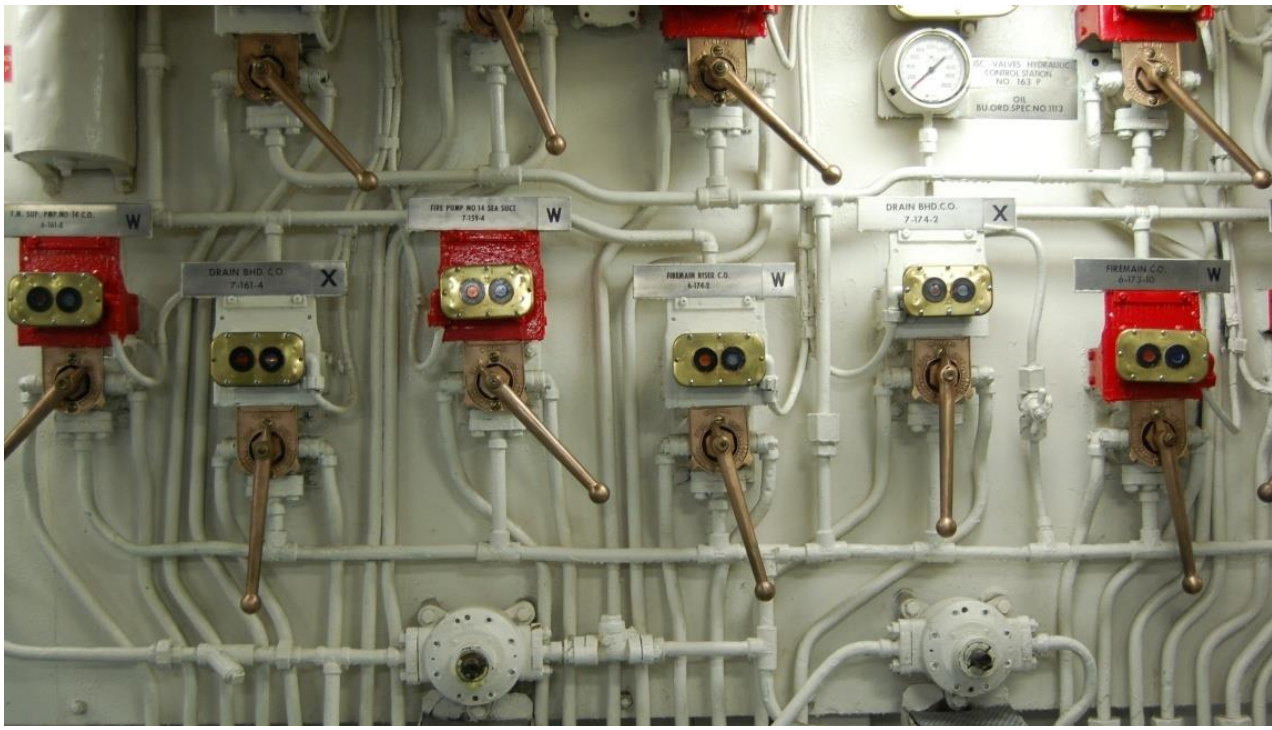
Summarize

28

29

Plumb

30



31

Program

32

33

Basics

34

Getting to the command line

- Windows: Cygwin + mintty or Windows Subsystem for Linux (WSL)
- Mac OS X: Applications – Utilities – Terminal
- Unix/Linux: terminal, xterm, rxvt, konsole, kvt, gnome-terminal, nxterm, eterm
- Android: Termux!


```
command
command <input-file
command >output-file
command1 | command2
command &
```

```
e=expansion
$e
$(command)
'literal string'
"string with \$ $e"
* ? [abc] [0-9]
```

37

38

```
command1 ; command2
(command1 ; command2)
$?
command1 || command2
command1 && command2
```



```
if command ; then
  command1
elif command ; then
   command2
else
   command3
fi
```

```
while command ; do
  commands
done

while read var ; do
  commands
done
```

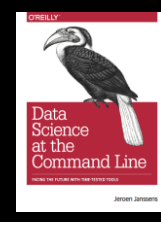


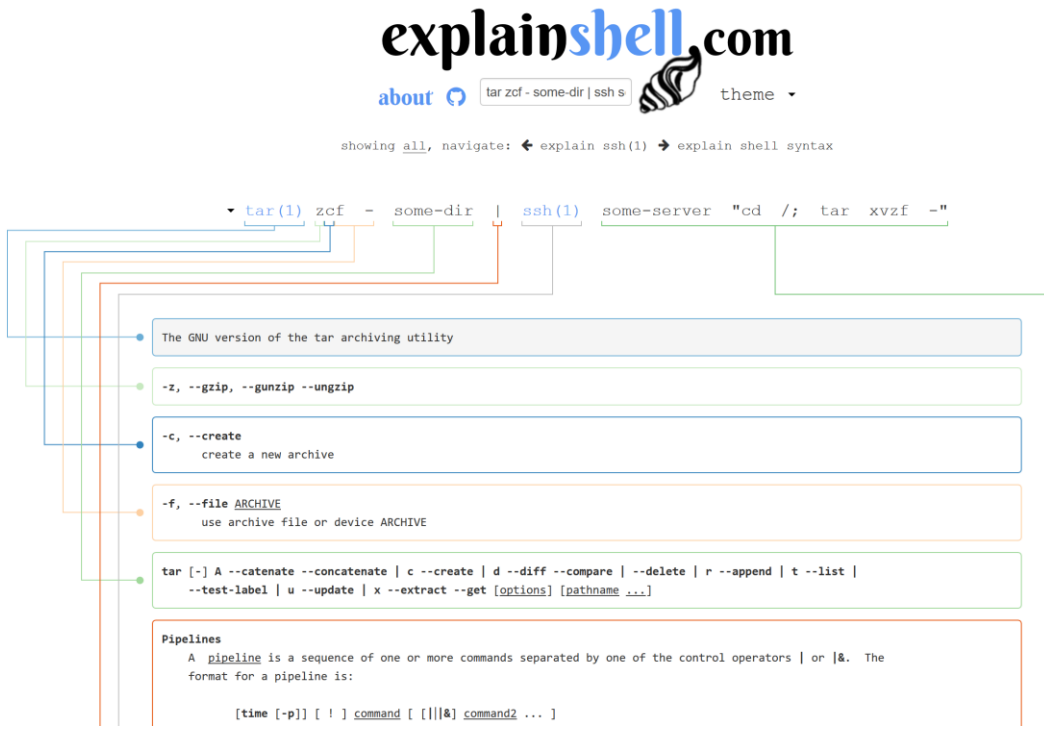
```
for var in a b c; do
  commands
done
```

```
case word in
pattern1) list ;;
pattern2) list ;;
esac
```


Help!

1. command --help
2. man command
3. GIYF
4. The Art of the Command Line
5. www.datascienceatthecommandline.com
6. explainshell.com
7. regex101.com




explainshell.com
about
tar zcf - some-dir | ssh s
theme
showing all, navigate: explain ssh(1) explain shell syntax
tar(1) zcf - some-dir | ssh(1) some-server "cd /; tar xvzf -"
The GNU version of the tar archiving utility
-z, --gzip, --gunzip --ungzip
-c, --create
create a new archive
-f, --file ARCHIVE
use archive file or device ARCHIVE
tar [-] A --catenate --concatenate | c --create | d --diff --compare | --delete | r --append | t --list |
--test-label | u --update | x --extract --get [options] [pathname ...]
Pipelines
A pipeline is a sequence of one or more commands separated by one of the control operators | or |&. The
format for a pipeline is:
[time [-p]] [!] command [[||&] command2 ...]

45

46

curl
wget

47

```
$ while read ticker
do
   curl \
"http://www.reuters.com/finan
ce/stocks/ratios?symbol=$tick
er&rpc=66#management" \
>$ticker.html
done <ticker_symbols
```

48

mysql
sqlplus
osql
sqlite3
psql
odbc


```
$ mysql -s db -e "select  distinct
substr(name, 1,length(name) -
 instr(reverse(name), '/'))
 from FILES" |
xargs ls -di |
awk '{print $1}' |
sort -u |
wc -l
```

ssh

51

```
$ ssh host.example.com \
  tar -cf - dir |
tar -xf -
```

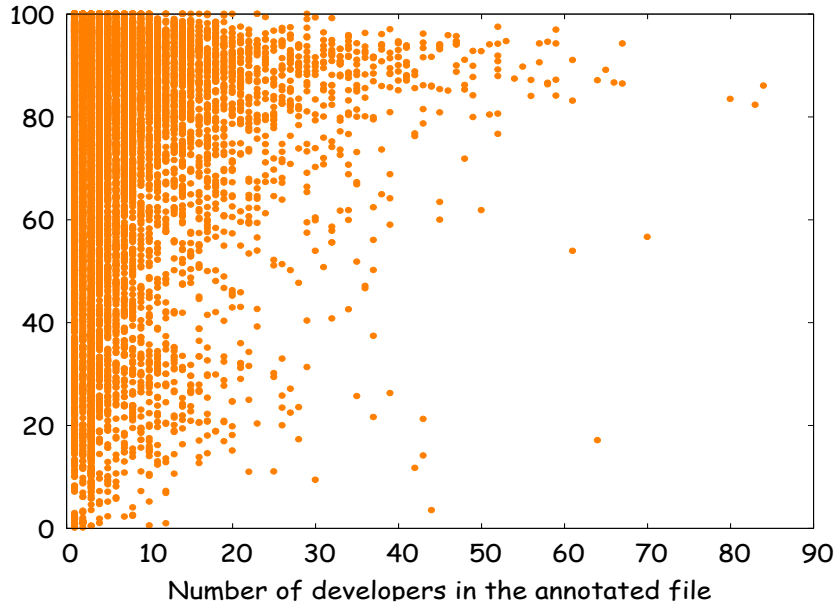
52

git log
git blame


```
$ git log remotes/origin/FreeBSD-release/11.1.0 -- usr.bin/sed/compile.c
commit d83c1ab3c4dec442d0635a08301d300ce8763139
Author: Pedro Giffuni <pfg@FreeBSD.org>
Date:   Tue Dec 16 20:26:11 2014 +0000

    sed: Bounds check the file path used in the 'w' command.

    Modified version of a diff from Sebastien Marie to prevent a crash found
    with the afl fuzzer.

    Obtained from:  OpenBSD (CVS Rev. 1.37)
    MFC after:      1 week

commit 940c50967ad952f1e10d556406f58f4d1725a20d
Author: Eitan Adler <eadler@FreeBSD.org>
Date:   Mon Dec 9 18:57:20 2013 +0000

    Per the resolution of POSIX bug 0000779 (note 0002050) add support for using 'i'
    as a case insensitive flag.

    PR:             standards/184641
    Requested by:   David A. Wheeler <dwheeler@dwheeler.com>
    MFC After:      1 week

commit 35cd6ad001e4b4b01c97eb3b0a76b458959d7cb2
Author: Diomidis Spinellis <dds@FreeBSD.org>
Date:   Sun Sep 20 15:47:31 2009 +0000

    IEEE Std 1003.1, 2004 Edition states:

    "The escape sequence '\n' shall match a <newline> embedded in
    the pattern space."

    It is unclear whether this also applies to a \n embedded in a
    character class.  Disable the existing handling of \n in a character
    class following Mac OS X, GNU sed version 4.1.5 with --posix, and
    SunOS 5.10 /usr/bin/sed.
```

```
$ git blame remotes/origin/FreeBSD-release/11.1.0 -- usr.bin/sed/compile.c
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  57)
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  58) #define LHSZ  128
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  59) #define       LHMASK  (LHSZ - 1)
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  60) static struct labhash {
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  61)       struct  labhash *lh_next;
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  62)       u_int   lh_hash;
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  63)       struct  s_command *lh_cmd;
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  64)       int     lh_ref;
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  65) } *labels[LHSZ];
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  66)
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  67) static char    *compile_addr(char *, struct s_addr *);
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  68) static char    *compile_ccl(char **, char *);
99c176f28d80b (Diomidis Spinellis  2009-09-20 15:17:40 +0000  69) static char    *compile_delimited(char *, char *, int);
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  70) static char    *compile_flags(char *, struct s_subst *);
9cbdc6ce9db91 (Suleiman Souhlal    2007-07-04 16:42:41 +0000  71) static regex_t         *compile_re(char *, int);
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  72) static char    *compile_subst(char *, struct s_subst *);
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  73) static char    *compile_text(void);
af5f6621bf570 (Tim J. Robbins      2004-07-14 10:06:22 +0000  74) static char    *compile_tr(char *, struct s_tr **);
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  75) static struct s_command
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  76)              **compile_stream(struct s_command **);
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  77) static char    *duptoeol(char *, const char *);
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  78) static void     enterlabel(struct s_command *);
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  79) static struct s_command
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  80)              *findlabel(char *);
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  81) static void     fixuplabel(struct s_command *, struct s_command
*);
5f25fb883c468 (Warner Losh         2002-03-22 01:42:45 +0000  82) static void     uselabel(void);
e718f31e7a9f5 (Rodney Grimes       1994-05-27 12:33:43 +0000  83)
```

55



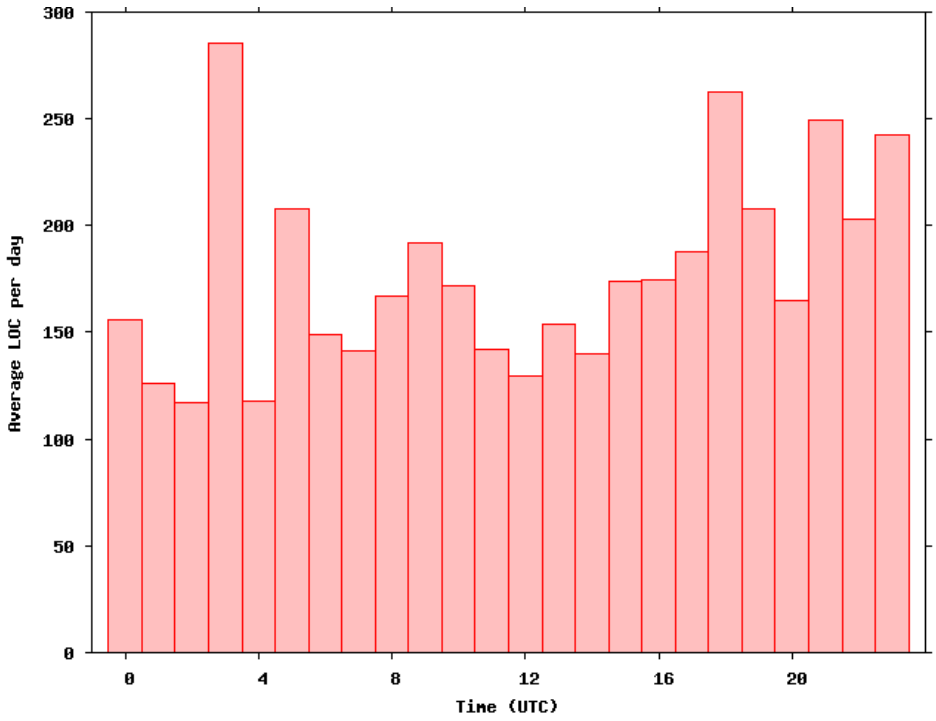
56

```
$ git blame -e f |
awk '{print $3}' |
sort |
uniq -c |
sort -rn
```

57

```
   1652 spy@...
   1104 polina
    827 ajk
    372 dds
    279 panos
    124 mastorad
     75 mm
     16 nd
      2 nmil
      1 pappas
```

58


Average LOC per day
300
250
200
150
100
50
0
0
4
8
12
16
20
Time (UTC)

59

find

60

```
find -name foo
 find -type f
 find -type d
 find -print0
```

61

```
$ find build \
 -name '*.class' \
 -type f \
 -mtime -7
```

62

nm,ldd,readelf
dumpbin
javap


```
$ for i in *.exe ; do
  echo -n "$i "
  dumpbin /imports $i |
  wc -l
done |
sort -k2nr |
head
```

```
mmc.exe 1300
mspaint.exe 1295
xpsrchvw.exe 1123
explorer.exe 1114
certutil.exe 888
vsgraphicsremoteengine.exe 772
icardagt.exe 696
osk.exe 677
eudcedit.exe 662
dxcap.exe 647
```

65

ar

ldd

66

```
$ cd /usr/bin
$ ldd * 2>/dev/null |
awk '/=>/{print $3}' |
sort |
uniq -c |sort -rn |
head
```

67

```
 372 /lib/libc.so.7
  35 /lib/libncurses.so.7
  33 /lib/libm.so.5
  31 /lib/libz.so.4
  29 /lib/libcrypt.so.4
  27 /lib/libutil.so.7
  27 /lib/libcrypto.so.5
  22 /usr/lib/libstdc++.so.6
  22 /lib/libgcc_s.so.1
  20 /usr/lib/libbz2.so.3
```

68

```
      541 /lib/libc.so.6
      541 /lib/ld-linux.so.2
      104 /lib/libdl.so.2
       92 /lib/libm.so.6
       62 /lib/libncurses.so.5
       59 /lib/libnsl.so.1
       50 /lib/libcrypt.so.1
       34 /usr/lib/libstdc++-ibc6.2-
2.so.3
       34 /lib/libpam.so.0
       28 /usr/lib/libz.so.1
```


tar

jar

```
$ tar -cf - . |
(cd dir ; tar -xpf -)
```

71

```
$ jar tvf /home/dds/hadoop-
2.5.0/share/hadoop/tools/lib/junit-
4.11.jar |
 head
     0 Wed Nov 14 20:21:28 EET 2012 META-INF/
   103 Wed Nov 14 20:21:26 EET 2012 META-INF/MANIFEST.MF
     0 Wed Nov 14 20:21:24 EET 2012 junit/
     0 Wed Nov 14 20:21:26 EET 2012 junit/extensions/
     0 Wed Nov 14 20:21:26 EET 2012 junit/framework/
     0 Wed Nov 14 20:21:26 EET 2012 junit/runner/
     0 Wed Nov 14 20:21:26 EET 2012 junit/textui/
     0 Mon Jul 09 20:49:42 EEST 2012 org/
     0 Wed Nov 14 20:21:26 EET 2012 org/junit/
     0 Wed Nov 14 20:21:24 EET 2012 org/junit/experimental/
```

72

readlog
winreg
docprop
winclip
odbc

www.spinellis.gr/sw/outwit/

perceval

```
usage: perceval [-c <file>] [-g] <backend> [<args>] | --help | --version

Repositories are reached using specific backends. The most common backends
are:
    askbot          Fetch questions and answers from Askbot site
    bugzilla        Fetch bugs from a Bugzilla server
    bugzillarest    Fetch bugs from a Bugzilla server (>=5.0) using its REST API
    confluence      Fetch contents from a Confluence server
    discourse       Fetch posts from Discourse site
    dockerhub       Fetch repository data from Docker Hub site
    gerrit          Fetch reviews from a Gerrit server
    git             Fetch commits from Git
    github          Fetch issues, pull requests and repository information from GitHub
    gitlab          Fetch issues, merge requests from GitLab
    googlehits      Fetch hits from Google API
    groupsio        Fetch messages from Groups.io
    hyperkitty      Fetch messages from a HyperKitty archiver
    jenkins         Fetch builds from a Jenkins server
    jira            Fetch issues from JIRA issue tracker
    launchpad       Fetch issues from Launchpad issue tracker
    mattermost      Fetch posts from a Mattermost server
    mbox            Fetch messages from MBox files
    mediawiki       Fetch pages and revisions from a MediaWiki site
    meetup          Fetch events from a Meetup group
    nntp            Fetch articles from a NNTP news group
    phabricator     Fetch tasks from a Phabricator site
    pipermail       Fetch messages from a Pipermail archiver
    redmine         Fetch issues from a Redmine server
    rss             Fetch entries from a RSS feed server
    slack           Fetch messages from a Slack channel
    stackexchange   Fetch questions from StackExchange sites
    supybot         Fetch messages from Supybot log files
    telegram        Fetch messages from the Telegram server
    twitter         Fetch tweets from the Twitter Search API
```

github.com/chaoss/grimoirelab-perceval

75

grep

76

```
a
.
^a
b$
k*
[a-z]
[^a-z]
\. \[ \*
(a.b) \1
a|b c+ d? {9} {,9}
```

77

```
$ egrep '[^aeiouyAEIOUY]{4}'\
/usr/share/dict/words
```

78

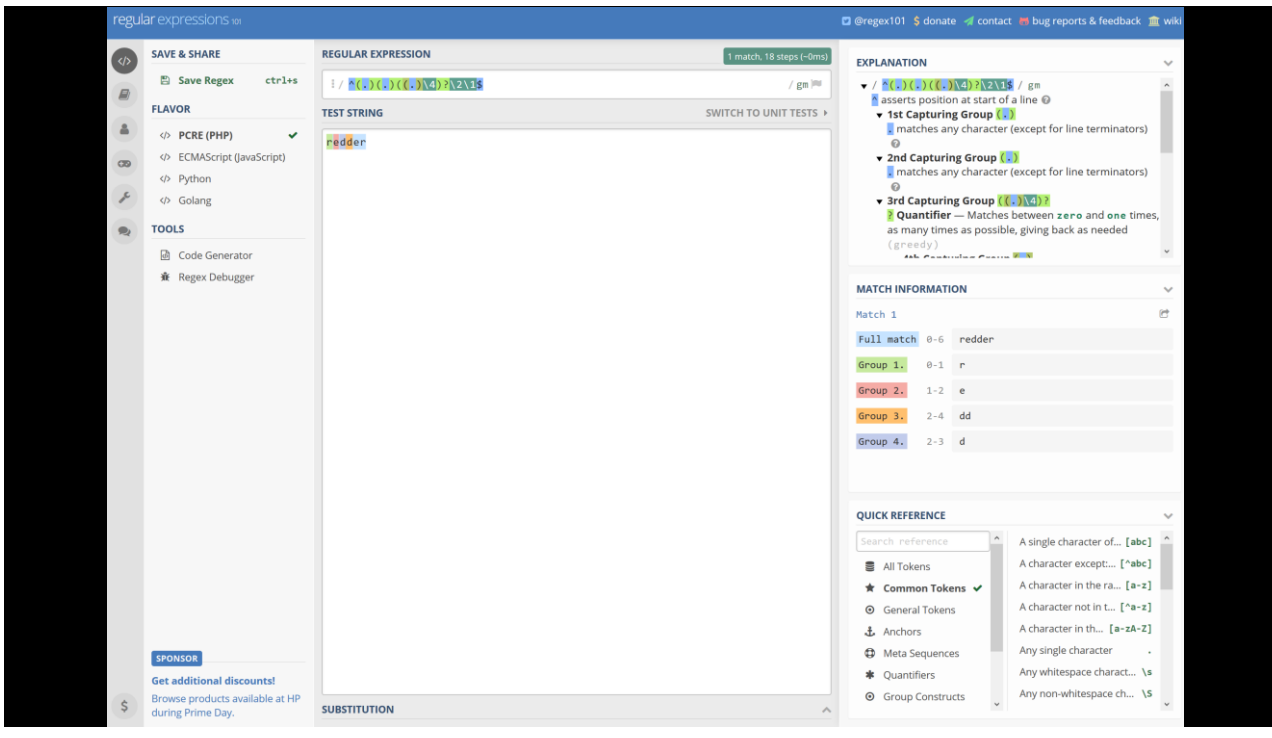
```
archchronicler
bergschrund
Eschscholtzia
fruchtschiefer
latchstring
lengthsman
Nachschlag
postphthisic
veldtschoen

Knightsbridge
```



```
$ egrep \
'^(.)(.)((.)\4)?\2\1$' \
/usr/share/dict/words
```

boob
deed
noon
peep
redder
sees

81



82

fgrep -f

awk

```
/^x/ {print $2}

$2 == $3 {a[$2]++}

rand() > 0.9

END {print x}

-F:
```



```
$ tar tvf vmmemctl.tar |
awk '{s += $3}
 END {print s}'
```

```
$ awk '!visited[$0]++'
```

87

sed

88

```
s/if(/if (/;s/do{/do {/

/^$/d

-n
s/^interface \(.*\) {/\1/p
```



```
$ ls
contact-de.html   faq-de.html       index-de.html
news-de.html      products-de.html  contact-en.html
faq-en.html       index-en.html     news-en.html
products-en.html  contact-fr.html   faq-fr.html
index-fr.html     news-fr.html      products-fr.html
$ mkdir en de fr
$ ls |
> sed -n 's/\([^-]*\)-\(..\)\.html/
> mv \1-\2.html \2\/\1.html/p'
```

cut

xml
xgawk

```
<xsl:param name="today"/>


$ xml tr report.xslt \
-s today=$(date +%Y0101)
\
file.xml >brochure.html
```


jq

```
$ curl -sLH "Accept: application/rdf+xml;q=0.5, application/vnd.citationstyles.c
sl+json;q=1.0" https://doi.org/10.1145/1391984.1391986 |jq .author
[
  {
    "given": "Panagiotis",
    "family": "Louridas",
    "sequence": "first",
    "affiliation": [
      {
        "name": "Athens University of Economics and Business, Athens, Greece"
      }
    ]
  },
  {
    "given": "Diomidis",
    "family": "Spinellis",
    "sequence": "additional",
    "affiliation": [
      {
        "name": "Athens University of Economics and Business, Athens, Greece"
      }
    ]
  },
  {
    "given": "Vasileios",
    "family": "Vlachos",
    "sequence": "additional",
    "affiliation": [
      {
        "name": "Athens University of Economics and Business, Athens, Greece"
      }
    ]
  }
] _
```

95

96

sort

97

```
$ tar tvf vmmemctl.tar  |
sort +2n
```

98

```
--numeric-sort -n
--reverse -r
--key=2nr -k2nr
--unique -u
--field-separator=: -t
--month-sort -m
```

99

comm

100

```
$ comm Linux FreeBSD
                [
arch
ash
bash
                                cat
                chflags
chgrp
                chio
                                chmod
chown
                                cp
cpio
                                csh
                                date
                                dd
                                df
dir
dmesg
dnsdomainname
                domainname
                                echo
                                ed
egrep
                expr
false
                getfacl
```

101

```
-1
-2
-3
```

102

```
$ cat mammals
cow
dog
dolphin
tiger
whale

$ cat sea-animals
dolphin
shark
trout
whale
```



```
# mammals ∪ sea-animals
$ sort -mu mammals sea-animals
cow
dog
dolphin
shark
tiger
trout
whale
```

```
$ # mammals ∩ sea-animals
$ comm -12 mammals sea-animals
dolphin
whale

$ # mammals - sea-animals
$ comm -23 mammals sea-animals
cow
dog
tiger
```

105

join

106

tsort

107

```
$ cat clothing
shirt necktie
socks shoes
shirt jacket
necktie vest
underwear shirt
underwear trousers
trousers shoes
trousers belt
belt necktie
shirt vest
vest jacket
jacket coat
trousers coat
```

```
$ tsort clothing
socks
underwear
trousers
shirt
belt
shoes
necktie
vest
jacket
coat
```

108

diff


```
$ diff file1 file2 |
grep '^[<>]' |
wc -l
```

```
--ignore-all-space
--unified
```

111

```
test
[
```

112

```
$ test -d $dir || mkdir $dir

if ! [ -r $file ] ; then
  echo "Unable to read $file" 1>&2
  exit 1
fi
```


expr

| & < <= = != > >=

\+ - * / %

: index substr length

tr

115

```
$ tr -C a-zA-Z '\n' </etc/motd |
tr A-Z a-z |
sort |
uniq |
comm -23 - /usr/share/dict/words
```

116

```
openssl enc -e -aes-256-cbc

openssl enc -d -aes-256-cbc
```

117

paste

118

119

120

```
$ rev /usr/share/dict/words |
> sort | rev
[…]
Archangelica
Melica
sicilica
silica
basilica
semisilica
jellica
majolica
plica
replica
```

121

date

122

```
$ date -u +'%Y-%m-%d %H:%M:%S')
2020-02-04 18:22:06

$ date -r update.sh  -I -u
2012-02-18
```

123

```
for TZ in America/Los_Angeles America/Chicago \
 America/New_York Europe/London Europe/Paris \
 Europe/Madrid Asia/Kolkata Asia/Shanghai Asia/Tokyo ; do
   echo -n "The time in $TZ is "
   date
done

The time in America/Los_Angeles is Tue, Feb  4, 2020 10:19:27
The time in America/Chicago is Tue, Feb  4, 2020 12:19:27
The time in America/New_York is Tue, Feb  4, 2020 13:19:27
The time in Europe/London is Tue, Feb  4, 2020 18:19:27
The time in Europe/Paris is Tue, Feb  4, 2020 19:19:27
The time in Europe/Madrid is Tue, Feb  4, 2020 19:19:28
The time in Asia/Kolkata is Tue, Feb  4, 2020 23:49:28
The time in Asia/Shanghai is Wed, Feb  5, 2020 02:19:28
The time in Asia/Tokyo is Wed, Feb  5, 2020 03:19:28
```

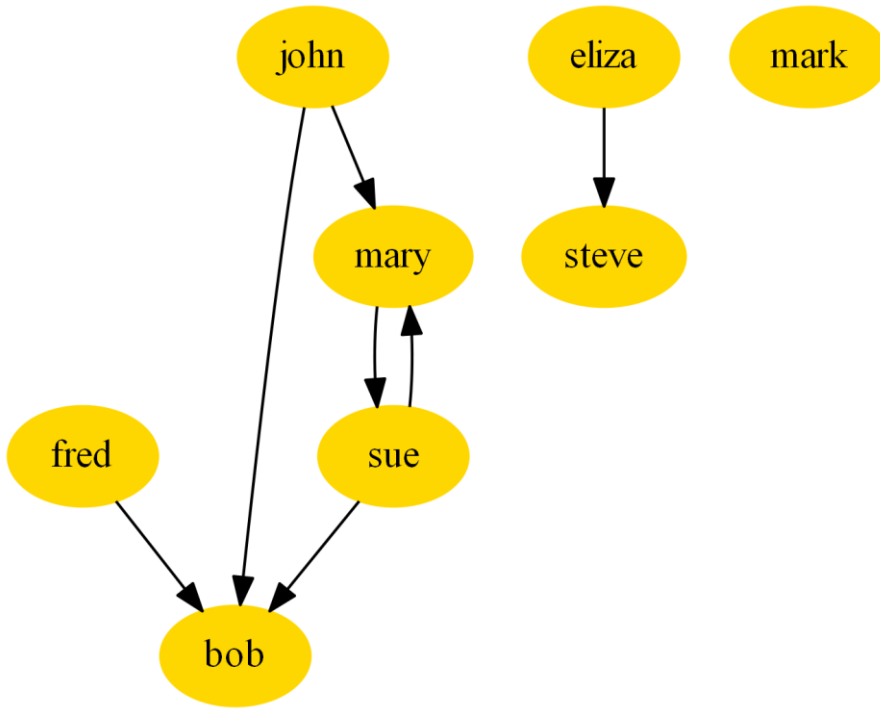
124

gvpr

125

```
digraph talks {
      bob;
      eliza;
      fred;
      john;
      mary;
      steve;
      sue;
      mark;

      john -> mary;
      john -> bob;
      mary -> sue;
      sue -> bob;
      sue -> mary;
      fred -> bob;
      eliza -> steve;
}
```

126



127

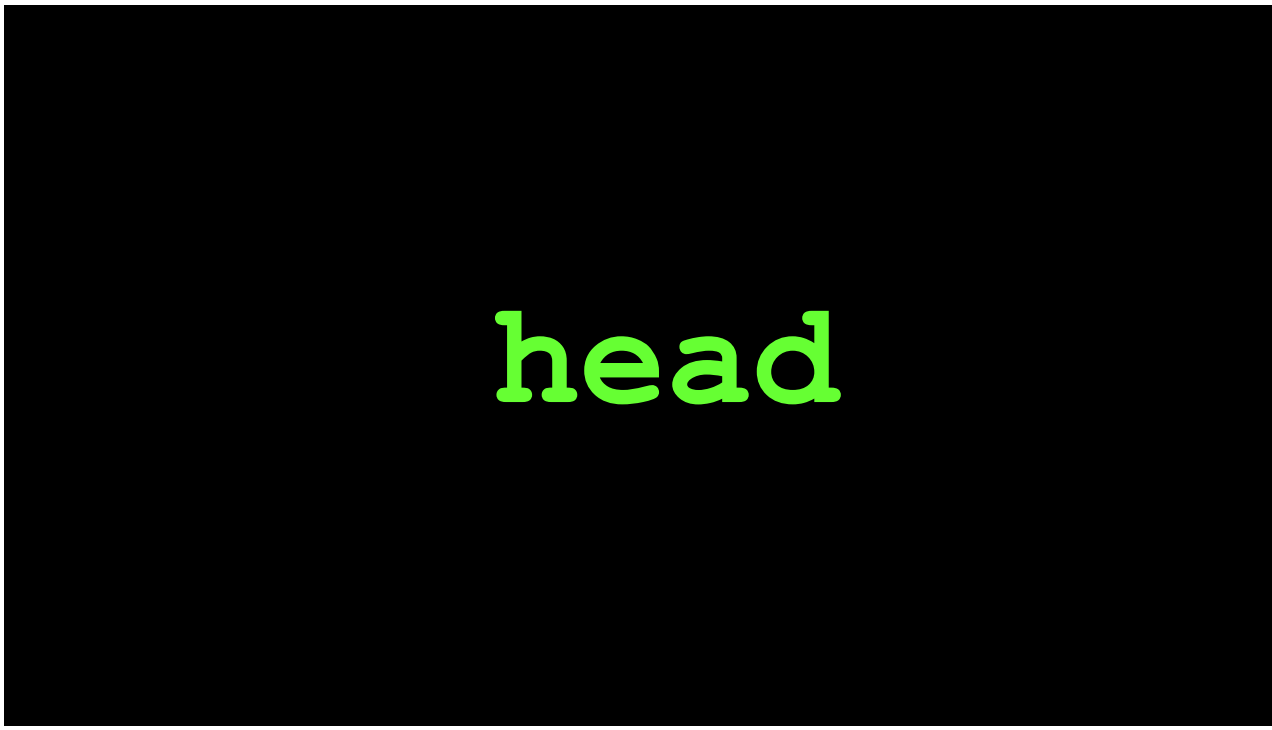
```
gvpr -c ' E {
  $.tail.talker = $.tail.talker + 1;
  $.head.listener = $.head.listener + 1
} ' <talk.dot
digraph talks {
      bob    [listener=3];
      eliza  [talker=1];
      steve  [listener=1];
      eliza -> steve;
      fred   [talker=1];
      fred -> bob;
      john   [talker=2];
      john -> bob;
      mary   [listener=2, talker=1];
      john -> mary;
      sue    [listener=1, talker=2];
      mary -> sue;
      sue -> bob;
      sue -> mary;
}
```

128

129

WC

130

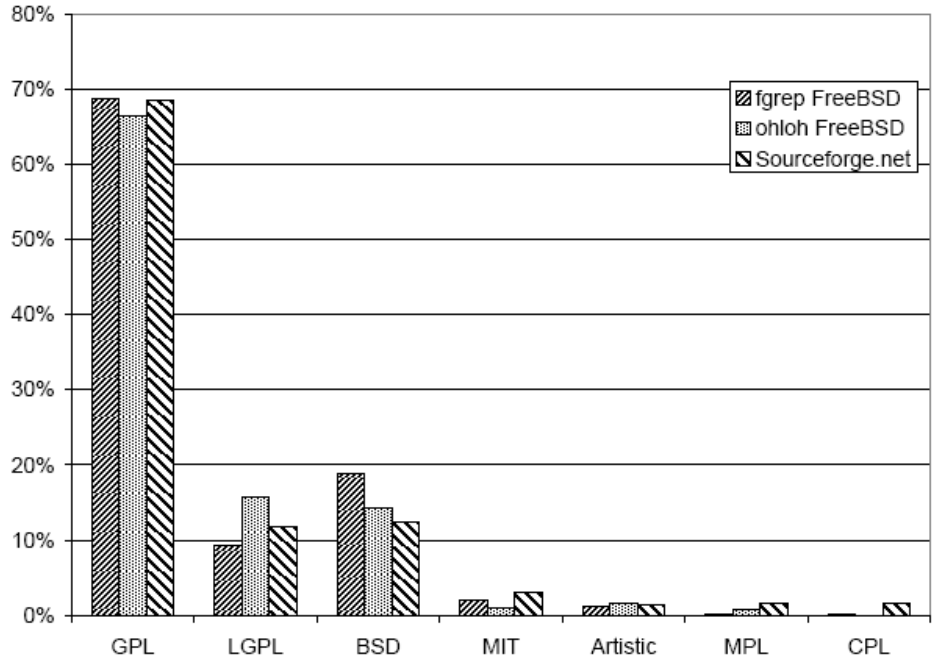


131

132

uniq


```
awk '{print $2}' licenses |
sort |
uniq -c |
sort -rn
```


80%
70%
60%
50%
40%
30%
20%
10%
0%
fgrep FreeBSD
ohloh FreeBSD
Sourceforge.net
GPL
LGPL
BSD
MIT
Artistic
MPL
CPL

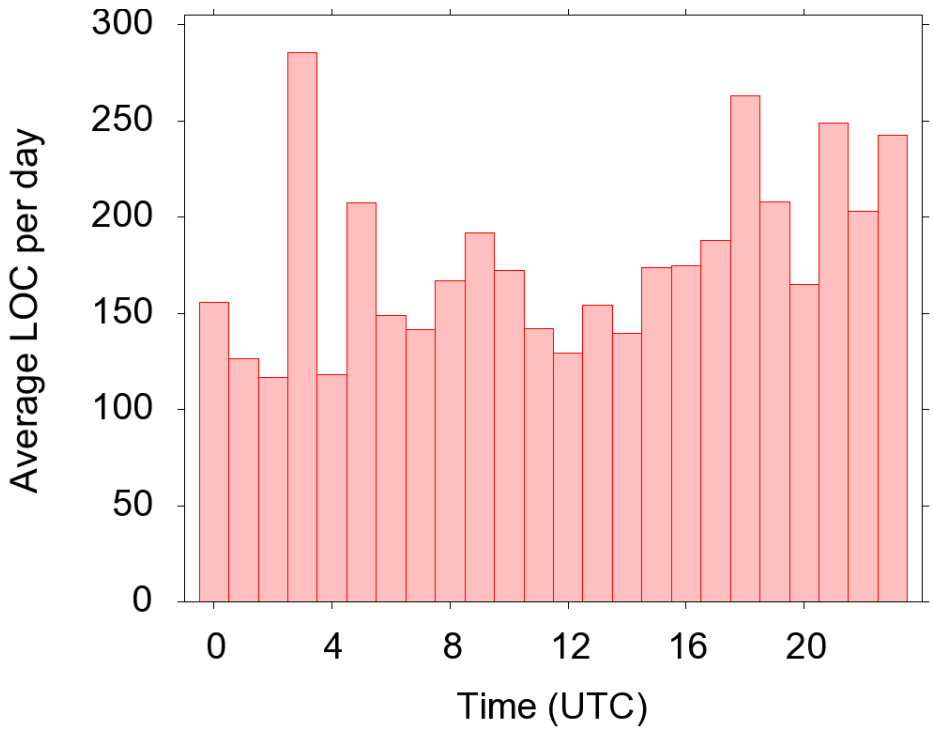
135

fmt

136

awk


```
find src -type f |
grep -v CVS |
xargs cvs log -SN |
sed -n '/^date:/s/[+;]//gp' |
awk '{hlines[$5] += $9}
 END {for (i in hlines)
       print i, hlines[i]}' |
sort >hlines
```

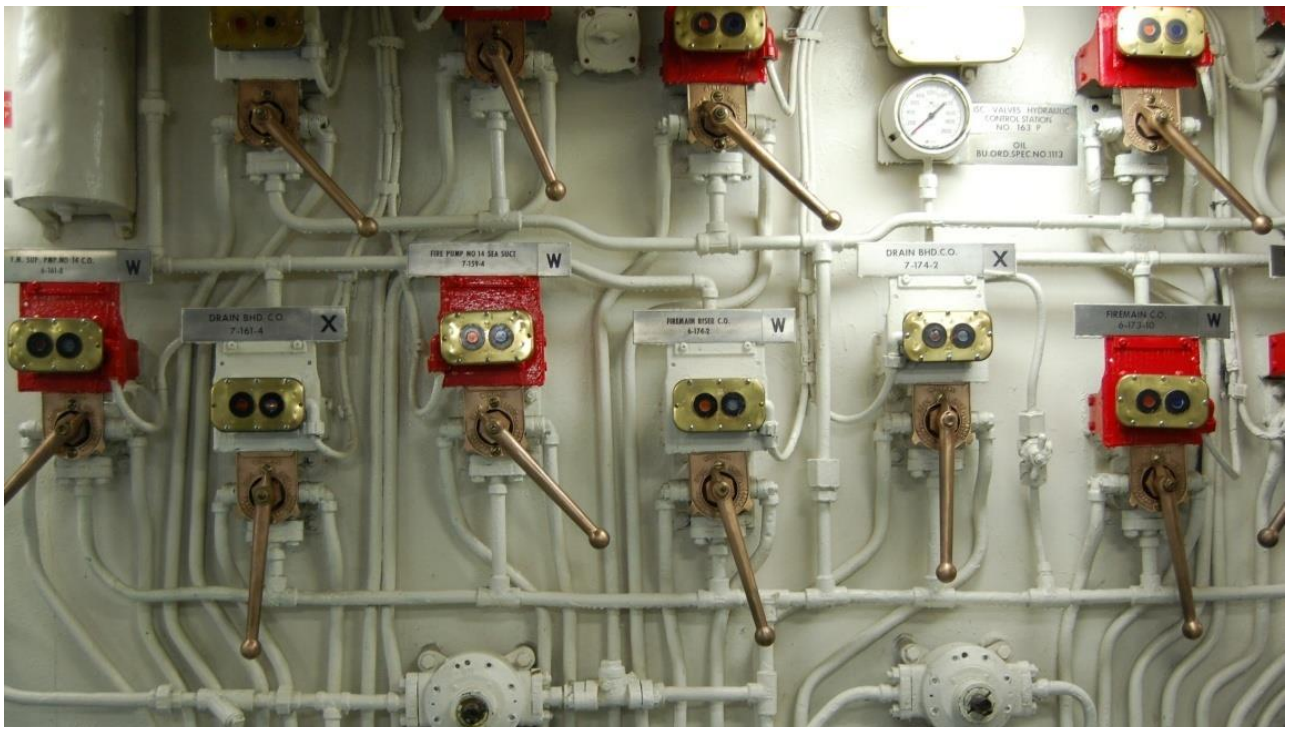


sendmail

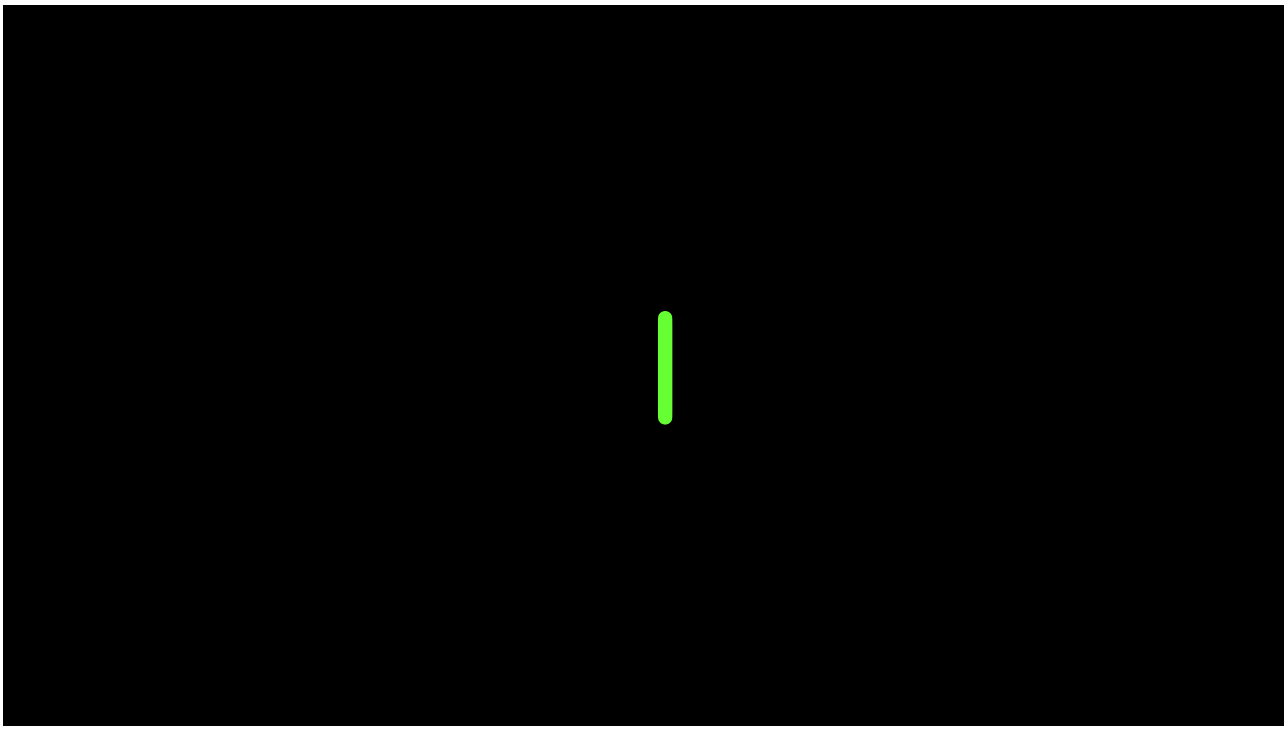
curl smtp:

```
du |
curl --ssl smtp://$HOST \
 --mail-from dds@aueb.gr \
 --mail-rcpt joe@example.com \
 --upload-file -
 --user "$USERNAME:$PASSWORD"
```

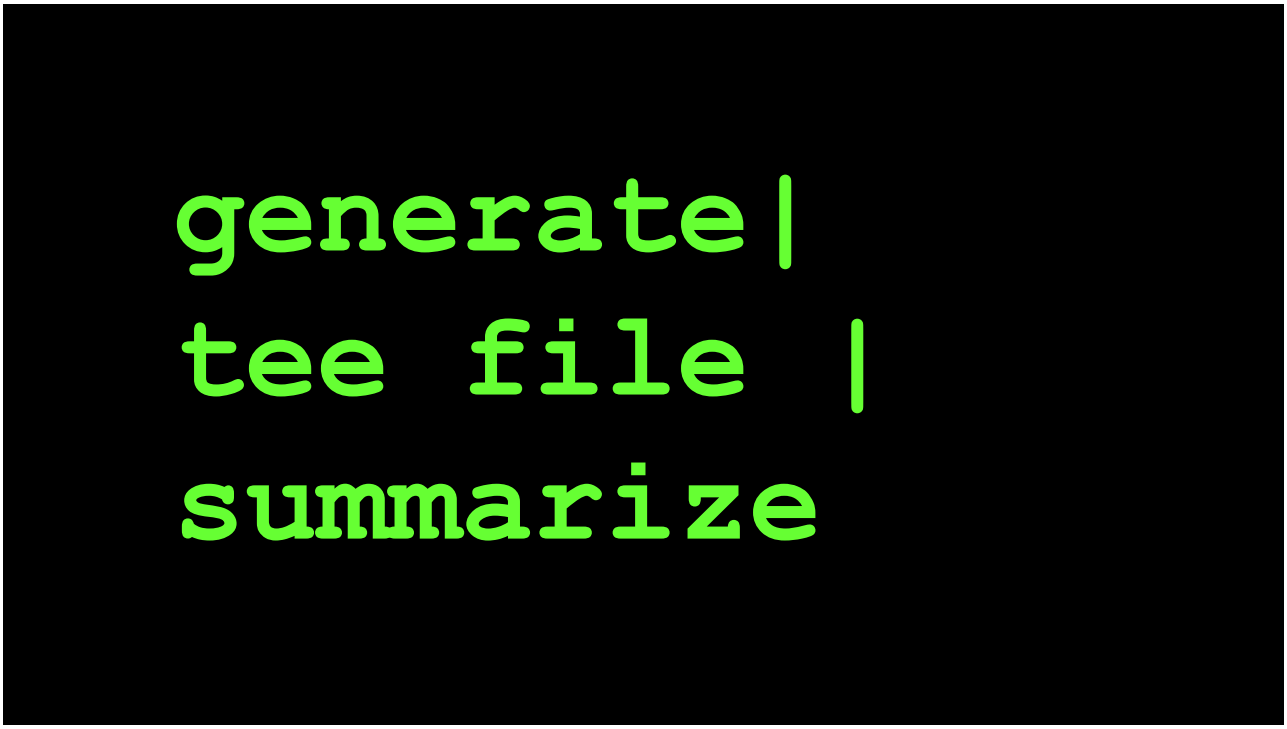
141



142



143



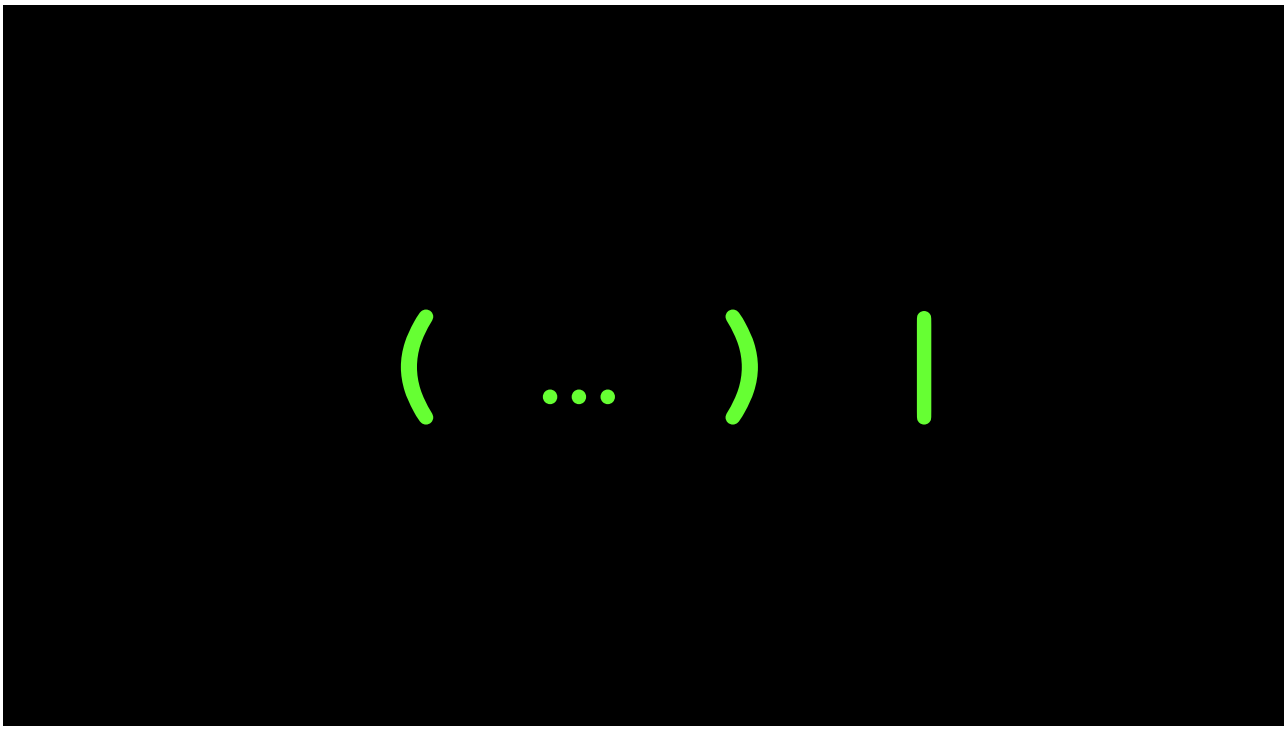
144

xargs

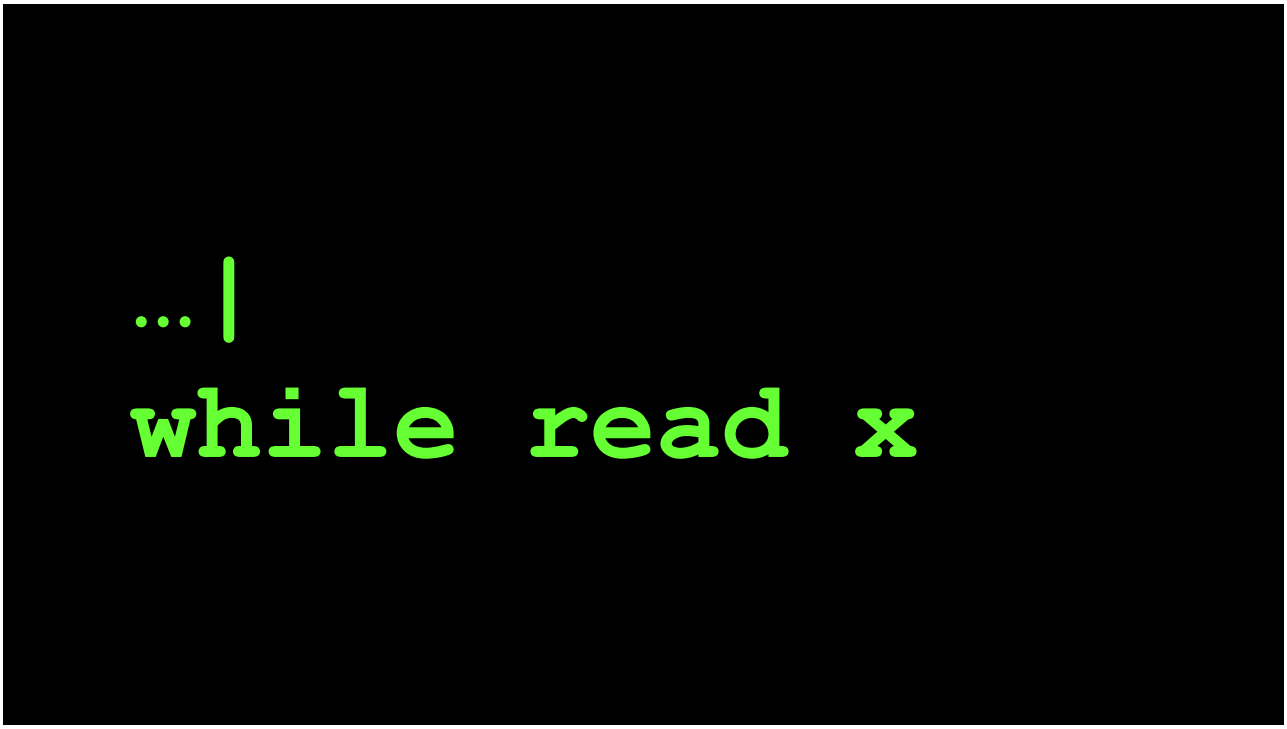
145

```
-0
-P
-n 1
```

146



147



148

```
|
if cmd
then
  …
fi |
```

149

```
<(…)
>(…)
$(…)
```

150

```
diff <(tar tvf a) <(tar tvf b)

command1 |
tee >(command2) |
tee >(command3) |
command4
```

151

152

Write a shell script

- Heavy lifting done by powerful tool (sort, grep, curl, git, sed, find, …)
- Script will glue diverse tools
- Workflow resembles a pipeline
- Steps can be interactively developed as shell commands
- Avoid dependency hell
- One-off job

Avoid shell scripting

- Difficult to see patterns on the left
- Hot loops
- Complex arithmetic / data structures / parameters / error handling
- Mostly binary data
- Large code body (> 500 LoC)
- Need a user interface

Java/Go/C/C++ < Python < sh < awk < sed


```
#!/bin/sh
…

chmod +x script-name

./script-name

mkdir $HOME/bin
mv script-name $HOME/bin
PATH="$PATH:$HOME/bin:."
```

Goodies

```
my_function() { … ; }

getopt

sh -x
```

155

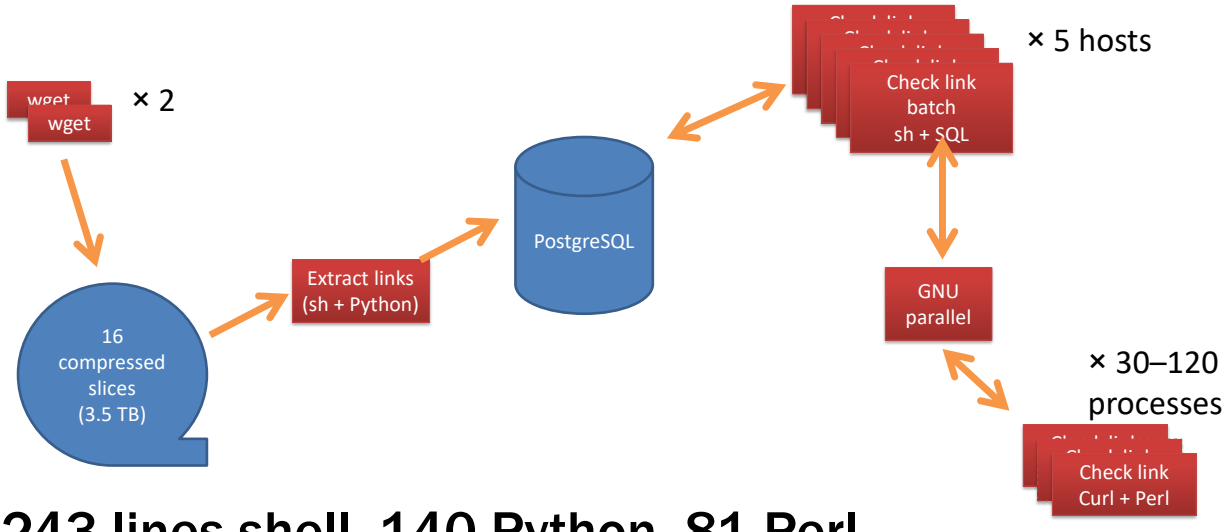
156

Key metrics

Documents: 71,158,279

Links: 18,552,379

Processing pipeline



243 lines shell, 140 Python, 81 Perl

```
#!/bin/bash
#
# Populate database from the contents of the specified slice
#

set -eu -o pipefail

SLICE=$1

ssh fetch-host "cd science &&
   bsdtar -xOf data/doc_ngrams_$SLICE.sql.zip |
   ./populate_db.py $SLICE" |
psql science_linking >/dev/null
```

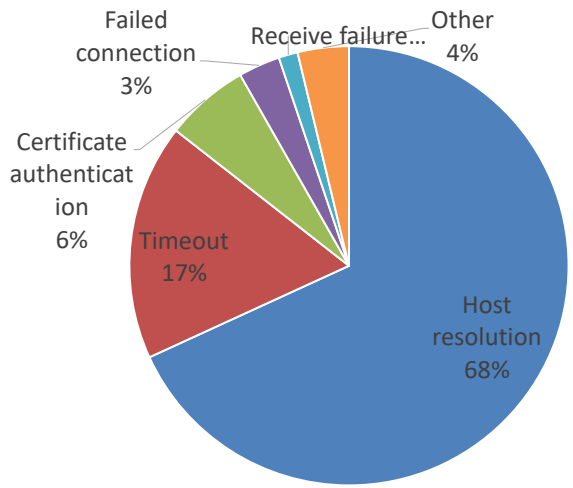
159

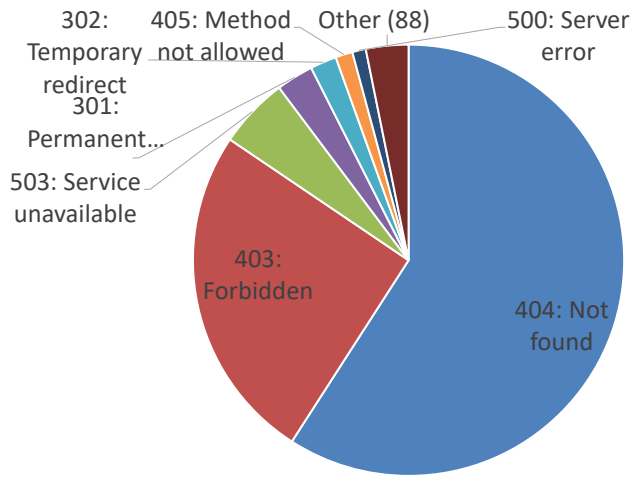
```
#!/bin/sh
#
# Check URI accessibility in parallel

{
  cat prepare-statement.sql
  echo "SELECT link_id,uri FROM link_checks …
        WHERE batch_id = '$BATCH_ID';" |
   ssh $DB_HOST psql --tuples-only … $DB |
   parallel --jobs 0 … ./check-url.sh {1} {2}
} |
ssh $DB_HOST psql $DB
```

160

```
#!/bin/sh
#
# Given a link-id and URL, check the URL and
# update the corresponding database record

LINK_ID="$1"
URL="$2"

ERRORS=/tmp/sl.$LINK_ID.err
HEADERS=/tmp/sl.$LINK_ID.out

curl --location --head --silent --max-time 60 --show-error \
  2>$ERRORS >$HEADERS "$URL"

echo "EXECUTE update_link_check('$LINK_ID', $(cat $ERRORS),"
perl header-to-json.pl "$HEADERS"
```


Network and HTTP failures





www.spinellis.gr
@CoolSWEng
dds@aueb.gr