

# curl and libcurl

[curl.haxx.se](http://curl.haxx.se)

# Agenda

- what's curl
- history
- what's libcurl
- development
- ownership
- users
- protocol
- license
- bindings
- alternatives
- future

Feel free to interrupt to ask questions!

# Daniel Stenberg

- Professional developer since 1991
- Consultant since 1996
- Contactor Data since 2000
  
- Open source-hacker: curl, Rockbox, roffit, Subversion, c-ares, libssh2, Dancer, FPL, wget, Hypermail, etc



CONTACTOR

# what's curl?

- command line tool
- get and send data using internet-protocols specified as a URL
- FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS and FILE
- “trust” the input
- unix-style (pipes, stdin/stdout, etc)
- shell front-end for libcurl

# cURL began

- what if I would write my own little program that would...
- httpget 1997 => urlget => curl 1998
- HTTP
- GOPHER
- FTP
- upload

# curl command line

- curl <http://curl.haxx.se/> -o storehere
- curl <ftp://curl.haxx.se/> -o storehere
- curl <https://curl.haxx.se/> -o storehere
- curl <sftp://curl.haxx.se/> -u me:pwd -o storehere
- curl <scp://curl.haxx.se/> -o storehere
- curl <tftp://curl.haxx.se/> -o storehere
- curl -T uploadfile <ftp://curl.haxx.se/tohere/>
- curl -T uploadfile <https://curl.haxx.se/tohere/>

# command line part 2

- 120 command line options!
- can be entered in a config file
- “man curl”
- curl --manual
- pipes, files, stdin/stdout

# libcurl began in 2000

- version 7
- API for other apps
- portable
- C
- PHP binding was early



# what's libcurl?

- functions for getting and sending data, given as URL for internet-related protocols
- written in C89 – for maximum portability, re-use, bindings etc
- doesn't care about the content
- offers blocking and non-blocking API (push vs pull)
- “hide” protocol-details when possible
- high performance
- do well at c10k+
- use thirds party libs where applicable

# why\_use\_libcurl?

- free, open and zero price
- thread-safe
- IPv6 compatible
- feature-rich
- well supported and maintained
- fast
- well documented
- portable

# libcurl\_hello\_world

- Using libcurl is very simple!

```
int main(void)
{
    CURL *curl;
    CURLcode result;

    curl = curl_easy_init();
    if(curl) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://curl.haxx.se");

        /* set more options here... */

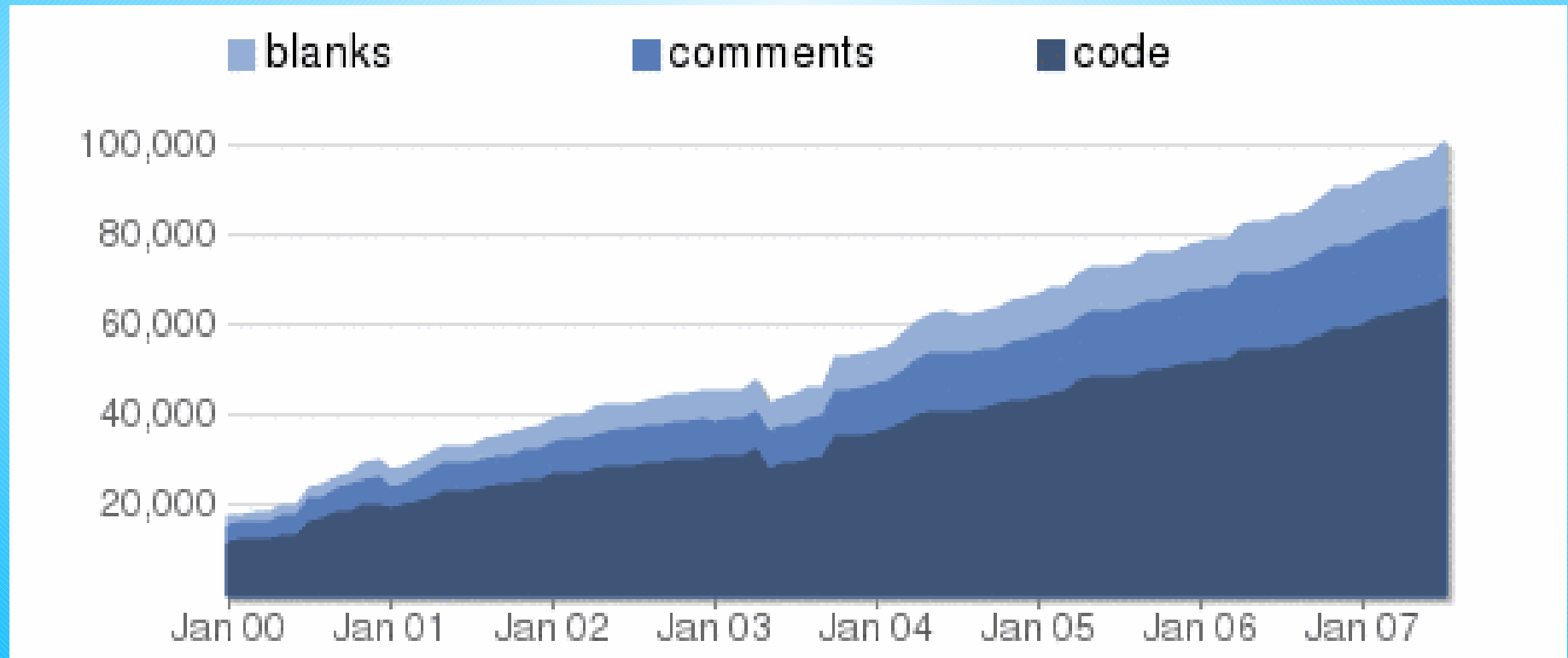
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
    return 0;
}
```

# contributors

- I started
- many provide “drive-by” patches
- most contributors fix a single problem
- few want to know nor care for “the big picture”
- a few sticks around, the core guys
- 10 persons have commit access
- ~600 named contributors so far

# over the years

- steady development
- scratch-an-itch



# my\_role

- Originator and project leader
- Spends roughly 2 hours/day
- Mostly mail support and patch reviewing
- Holding the wheel, saying what I like and not and what's good or bad.

# license

- started as GPL
- turned MPL oct 1998
- changed to MIT/MPL jan 2001
- changed to MIT sep 2002
- will not change license again, ever

# who owns curl?

- Daniel Stenberg owns the copyright
- Daniel does > 90% of all commits
- Almost all larger/major changes were done by D
- Assumes that people own or have rights to their contributions
- Carefully attributes and gives credit to contributors for their efforts

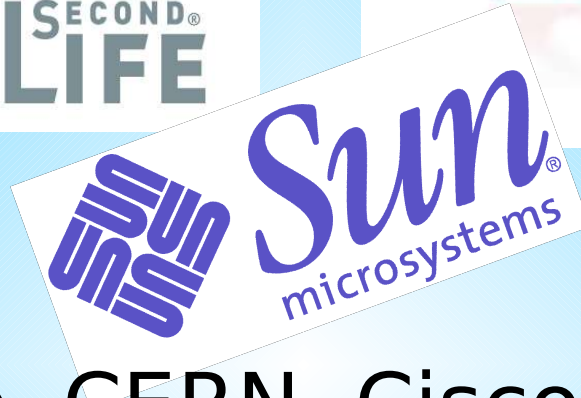
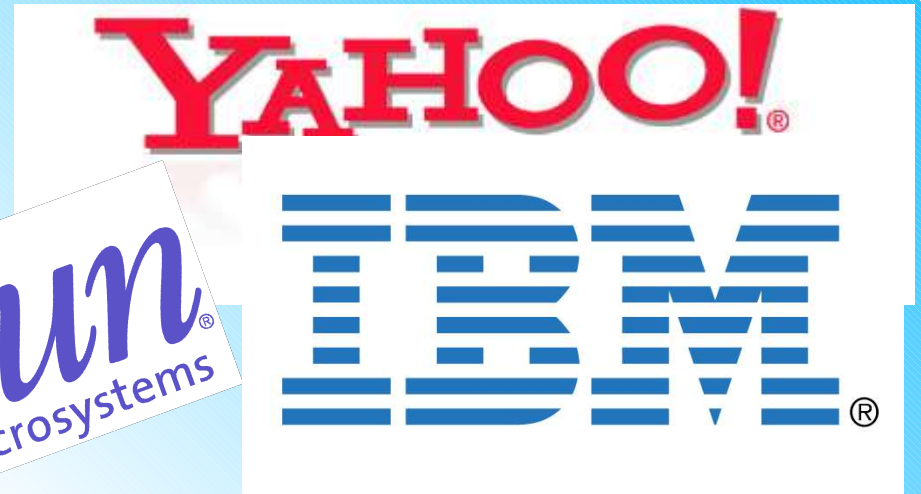


# OS- and Linux- distros

- Linux: Red Hat, Fedora, Suse, Debian, Mandriva, Ubuntu, Gentoo and many more
- Others: Mac OS X, IBM AIX, SGI IRIX, Sun Solaris, FreeBSD, OpenBSD, NetBSD, UnixWare, HP-UX
- they don't tell us
- don't contribute in any particular extent
- “absorbs bug reports” for good and bad
- lagging behind terribly at times
- not always good at telling where things come from (= they didn't create this thing)

# commercial use

- license
- portability
- stability
- Adobe, Apple, CERN, Cisco, Google, F-Secure, IBM, Linden Lab, MacAfee, Polaroid, RSA Inc, SAS Institute, SEB, Siemens, Sun, Symantec, Yahoo, etc



# open source use

- GPL compatible
- Exists in “all” distros
- Well documented
- OpenOffice.org, git, Firefox3, Gnash, gimp, libtorrent, Boinc, Xen, Gizmo, metadl, curl-loader, grip, UnrealIRCd, Aegis, Cmake, slapt-get, Xine, AlsaPlayer, xmms2, EmBrowser, OpenVRML, libcompres and more

# how many users?

- >1 million downloads per year
- distros offer their own packages
- comes on “bonus”-CDs
- comes with applications
- simply no way to measure

# portability

- Solaris, NetBSD, FreeBSD, OpenBSD, Darwin, HPUX, IRIX, AIX, Tru64, Linux, UnixWare, HURD, Windows, Amiga, OS/2, BeOs, Mac OS X, Ultrix, QNX, OpenVMS, RISC OS, Novell NetWare, DOS, MVS, VxWorks, Windows CE, OS/400 ...
- Alpha, HP-PA, HP3000, MIPS, PowerPC, POWER, SuperH4, Sparc, ARM, i386, ia64, m68k, s390, ...
- no other transfer-lib is even close
- autobuilds and tests
- still relies on users to report problems

# test

- Test against requirement?
- Autobuilds
- CVS-daily snapshots
- ask on the mailing list
- no formal
- no unit-tests...

# protocols

- HTTP, FTP, TELNET, TFTP, FILE, DICT
- HTTPS, FTPS - SSL-libraries (OpenSSL, GnuTLS, NSS, yassl, oqssl, ...)
- LDAP - openldap
- SCP, SFTP - libssh2
- libcurl is standards compliant, but adjusts to what common browsers do

# community

- users / library mailing lists (100 / 300 mails/month in average)
- bugs, bugtracking
- new ideas
- design discussions
- post patches!
- bug and feature-request trackers
- *everyone can join!*



# who pays?

- my family
- volunteers
- many use/fix bugs on paid work time or for commercial projects
- a few companies have sponsored/paid development of certain features etc
- The Swedish IIS fund 2005/2006

# bindings

- Ada95, Basic, C, C++, Ch, Cocoa, D, Dylan, Euphoria, Ferite, Gambas, glib/GTK+, Java, Lisp, Lua, Mono, .NET, Object-Pascal, Ocaml, Pascal, Perl, PHP, Postgres, Python, R, Rexx, Ruby, Scheme, S-Lang, Smalltalk, SPL, Tcl, Visual Basic, Q, wxWidgets, XBLite, and more!
- all major environments covered
- maintained outside of the cURL project

# libcurl alternatives

- C? libwww, neon, libsoup – not as portable, less functions, fewer protocols
- language-specific ones are common, but then too mostly not as feature-rich
- no idea about commercial ones, anyone?

# curl vs wget



## curl

- libcurl
- pipes
- return codes
- single-shot
- many protocols
- more portable
- more SSL libraries
- more HTTP auth
- bidirectional

## wget

- command line
- recursive fetch
- older
- much less developer activity
- HTTP 1.0
- GPL
- GNU

# curl numbers, nov 2007



- number of curl releases: 102
- unique web site visitors/month: 110,000+
- web site mirrors: 43
- number of contributors: 600
- mails in user + library lists (jan-oct 2007)  
5969
- mails written by Daniel: 1460
- comment ratio in code: 24%
- number of code lines: 68K - 110K

# the future!

- faster and better
- more protocols
- greater control over details
- fewer irrelevant functions
- more and better tests
- used more widely
- depends on what people want **and do**

Questions?

visit us at

[curl.haxx.se](http://curl.haxx.se)