

libcurl in the Real World

Dan Fandrich

May 2026
curl://up 2026



Using libcurl

Using libcurl

- Designed to satisfy many applications

Using libcurl

- Designed to satisfy many applications
- Large API to make everyone happy

Using libcurl

- Designed to satisfy many applications
- Large API to make everyone happy
 - 100 functions

Using libcurl

- Designed to satisfy many applications
- Large API to make everyone happy
 - 100 functions
 - 351 CURLOPT_ options

Using libcurl

- Designed to satisfy many applications
- Large API to make everyone happy
 - 100 functions
 - 351 CURLOPT_ options
 - 123 CURLE_ errors

Using libcurl

- Designed to satisfy many applications
- Large API to make everyone happy
 - 100 functions
 - 351 CURLOPT_ options
 - 123 CURLE_ errors
 - Many more: 1270 symbols in total

Using libcurl

Using libcurl

- No application uses it all

Using libcurl

- No application uses it all
- But what parts *do* they use?

Using libcurl

- No application uses it all
- But what parts *do* they use?
- Many are optional

Using libcurl

- No application uses it all
- But what parts *do* they use?
- Many are optional
- Some are mutually exclusive or deprecated
 - `curl_form*` vs `curl_mime*`
 - `curl_multi_socket` vs `curl_multi_socket_action`
 - `CURLOPT_PROGRESSFUNCTION` vs `CURLOPT_XFERINFOFUNCTION`
 - `CURLE_OPERATION_TIMEOUTED` vs `CURLE_OPERATION_TIMEDOUT`

Changing libcurl

Changing libcurl

- libcurl tries to be highly backward-compatible

Changing libcurl

- libcurl tries to be highly backward-compatible
- Bug fixes are actually behaviour changes

Changing libcurl

- libcurl tries to be highly backward-compatible
- Bug fixes are actually behaviour changes
- Sometimes necessary to

Changing libcurl

- libcurl tries to be highly backward-compatible
- Bug fixes are actually behaviour changes
- Sometimes necessary to
 - Deprecate APIs

Changing libcurl

- libcurl tries to be highly backward-compatible
- Bug fixes are actually behaviour changes
- Sometimes necessary to
 - Deprecate APIs
 - Change return codes

Changing libcurl

- libcurl tries to be highly backward-compatible
- Bug fixes are actually behaviour changes
- Sometimes necessary to
 - Deprecate APIs
 - Change return codes
 - Change behaviour

What parts are being used?

What parts are being used?

- How can curl developers know what parts are being used?

What parts are being used?

- How can curl developers know what parts are being used?
- Annual survey gives clues

What parts are being used?

- How can curl developers know what parts are being used?
- Annual survey gives clues
- Tracking people's issues, feature requests, pull requests

What parts are being used?

What parts are being used?

- libcurl is Open Source

What parts are being used?

- libcurl is Open Source
- Many libcurl-using applications are as well

What parts are being used?

- libcurl is Open Source
- Many libcurl-using applications are as well
- Why not just look?

What parts are being used?

- libcurl is Open Source
- Many libcurl-using applications are as well
- Why not just look?
- Just download the source and see

Finding Applications

Finding Applications

- Just Google it

Finding Applications

- Just Google it
- Manual, doesn't scale

Finding Applications

- Just Google it
- Manual, doesn't scale
- Need to manually download hundreds/thousands of apps

Finding Applications

- Just Google it
- Manual, doesn't scale
- Need to manually download hundreds/thousands of apps
- Distributions have done the hard work for us

Finding Applications

- Just Google it
- Manual, doesn't scale
- Need to manually download hundreds/thousands of apps
- Distributions have done the hard work for us
- Limited to C/C++ applications using libcurl directly

Choosing a Distro

Choosing a Distro

- Restrict search to Linux only for maximum reach

Choosing a Distro

- Restrict search to Linux only for maximum reach
- Distrowatch lists 1201 distros

Choosing a Distro

- Restrict search to Linux only for maximum reach
- Distrowatch lists 1201 distros
- Big one to get as many apps as possible

Choosing a Distro

- Restrict search to Linux only for maximum reach
- Distrowatch lists 1201 distros
- Big one to get as many apps as possible
- Fairly recent to get up-to-date versions

Choosing a Distro

- Restrict search to Linux only for maximum reach
- Distrowatch lists 1201 distros
- Big one to get as many apps as possible
- Fairly recent to get up-to-date versions
- Easy to find just libcurl-using applications

Choosing a Distro

- Restrict search to Linux only for maximum reach
- Distrowatch lists 1201 distros
- Big one to get as many apps as possible
- Fairly recent to get up-to-date versions
- Easy to find just libcurl-using applications
- Easy to download the source code

Choosing a Distro

Choosing a Distro

- Arch: 409

Choosing a Distro

- Arch: 409
- Debian Testing: 446

Choosing a Distro

- Arch: 409
- Debian Testing: 446
- Fedora 43: 397

Choosing a Distro

- Arch: 409
- Debian Testing: 446
- Fedora 43: 397
- Fedora 43 w/RPM Fusion: 419

Choosing a Distro

- Arch: 409
- Debian Testing: 446
- Fedora 43: 397
- Fedora 43 w/RPM Fusion: 419
- nixOS: 253

Choosing a Distro

- Arch: 409
- Debian Testing: 446
- Fedora 43: 397
- Fedora 43 w/RPM Fusion: 419
- nixOS: 253
- openSUSE Tumbleweed: 306

Choosing a Distro

Choosing a Distro

- Settled on Debian Testing

Choosing a Distro

- Settled on Debian Testing
- 441 rdepends packages

Choosing a Distro

- Settled on Debian Testing
- 441 rdepends packages
- 367 individual source tarballs

Choosing a Distro

- Settled on Debian Testing
- 441 rdepends packages
- 367 individual source tarballs
- 28.5 GiB source code

Choosing a Distro

- Settled on Debian Testing
- 441 rdepends packages
- 367 individual source tarballs
- 28.5 GiB source code
- 1,710,001 source files

Type of Analysis

Type of Analysis

- Run-time
 - Limited to what features we can figure out
 - Not scalable

Type of Analysis

- Run-time
 - Limited to what features we can figure out
 - Not scalable
- Static binary
 - nm on the executable
 - Skips use of options, error codes, macros, etc.

Type of Analysis

- Run-time
 - Limited to what features we can figure out
 - Not scalable
- Static binary
 - nm on the executable
 - Skips use of options, error codes, macros, etc.
- Compile of source
 - Too difficult with many different applications
 - Would need tooling to extract the data we want

Type of Analysis

Type of Analysis

- Static source
 - Most flexible
 - Access to all features used by the program
 - Access to all build configurations

Type of Analysis

- Static source
 - Most flexible
 - Access to all features used by the program
 - Access to all build configurations
- Cons:
 - Misses obfuscated symbols, e.g. macros wrapping `CURLOPT_*`
 - Simplistic scanning means some references in e.g. docs count
 - Can include source files using non-C bindings

Processing

Processing

- Some vendored libcurl sources found embedded in packages

Processing

- Some vendored libcurl sources found embedded in packages
- Some copies of `curl.h` found

Processing

- Some vendored libcurl sources found embedded in packages
- Some copies of `curl.h` found
- Some copies of `symbols-in-versions` found

Processing

- Some vendored libcurl sources found embedded in packages
- Some copies of `curl.h` found
- Some copies of `symbols-in-versions` found
- Removed (mostly) to avoid diluting results

Processing

Processing

- Extract all functions & symbols from curl source code

Processing

- Extract all functions & symbols from curl source code
- Files like `symbols-in-versions`, `test1135`, `VERSIONS.md` made it easy(ish)

Processing

- Extract all functions & symbols from curl source code
- Files like `symbols-in-versions`, `test1135`, `VERSIONS.md` made it easy(ish)
- Supplanted with version info from man page `.md` files

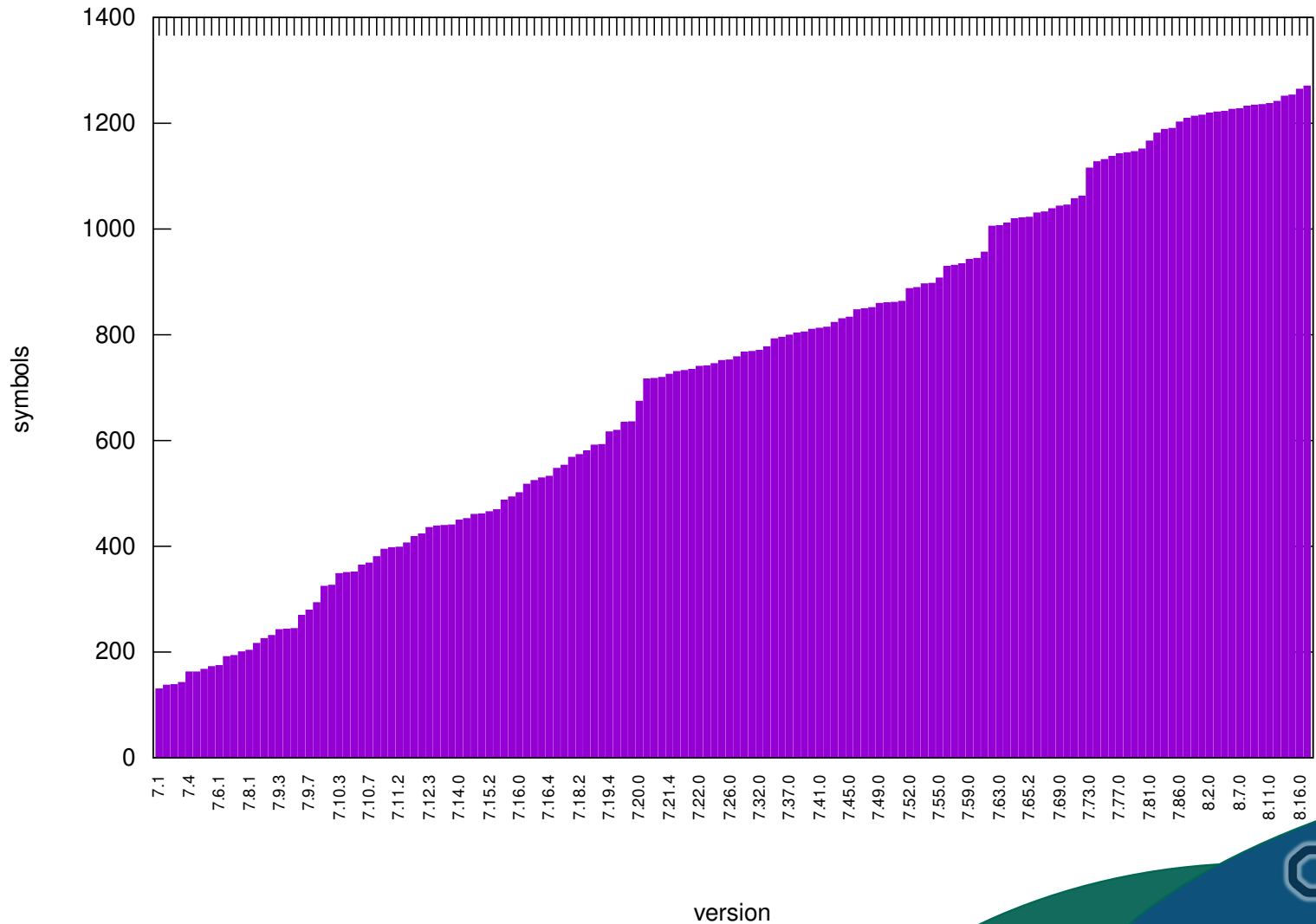
Processing

- Extract all functions & symbols from curl source code
- Files like `symbols-in-versions`, `test1135`, `VERSIONS.md` made it easy(ish)
- Supplanted with version info from man page `.md` files
- Massive grep of all symbols over all source code

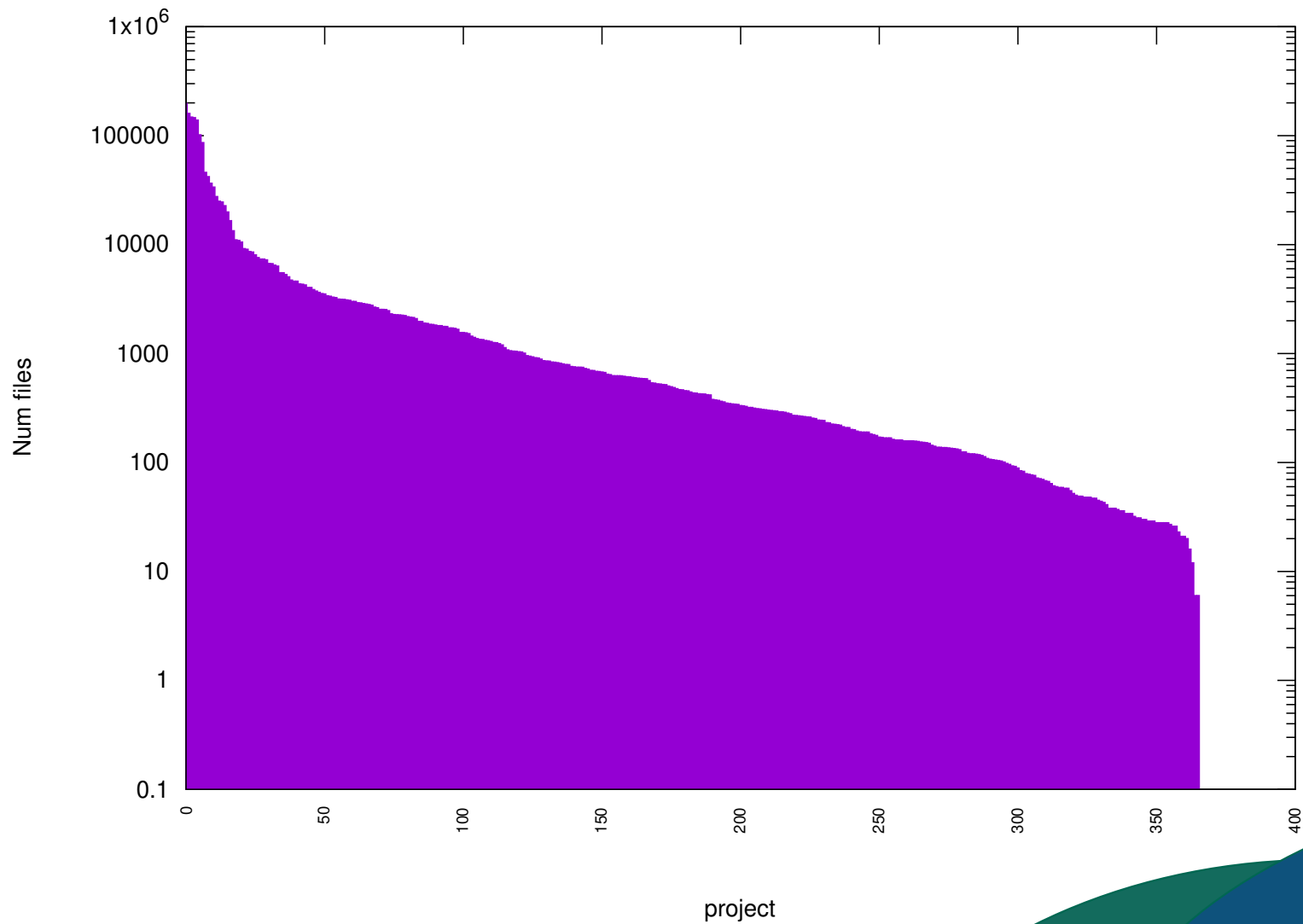
Processing

- Extract all functions & symbols from curl source code
- Files like `symbols-in-versions`, `test1135`, `VERSIONS.md` made it easy(ish)
- Supplanted with version info from man page `.md` files
- Massive grep of all symbols over all source code
- Results entered into sqlite3 DB

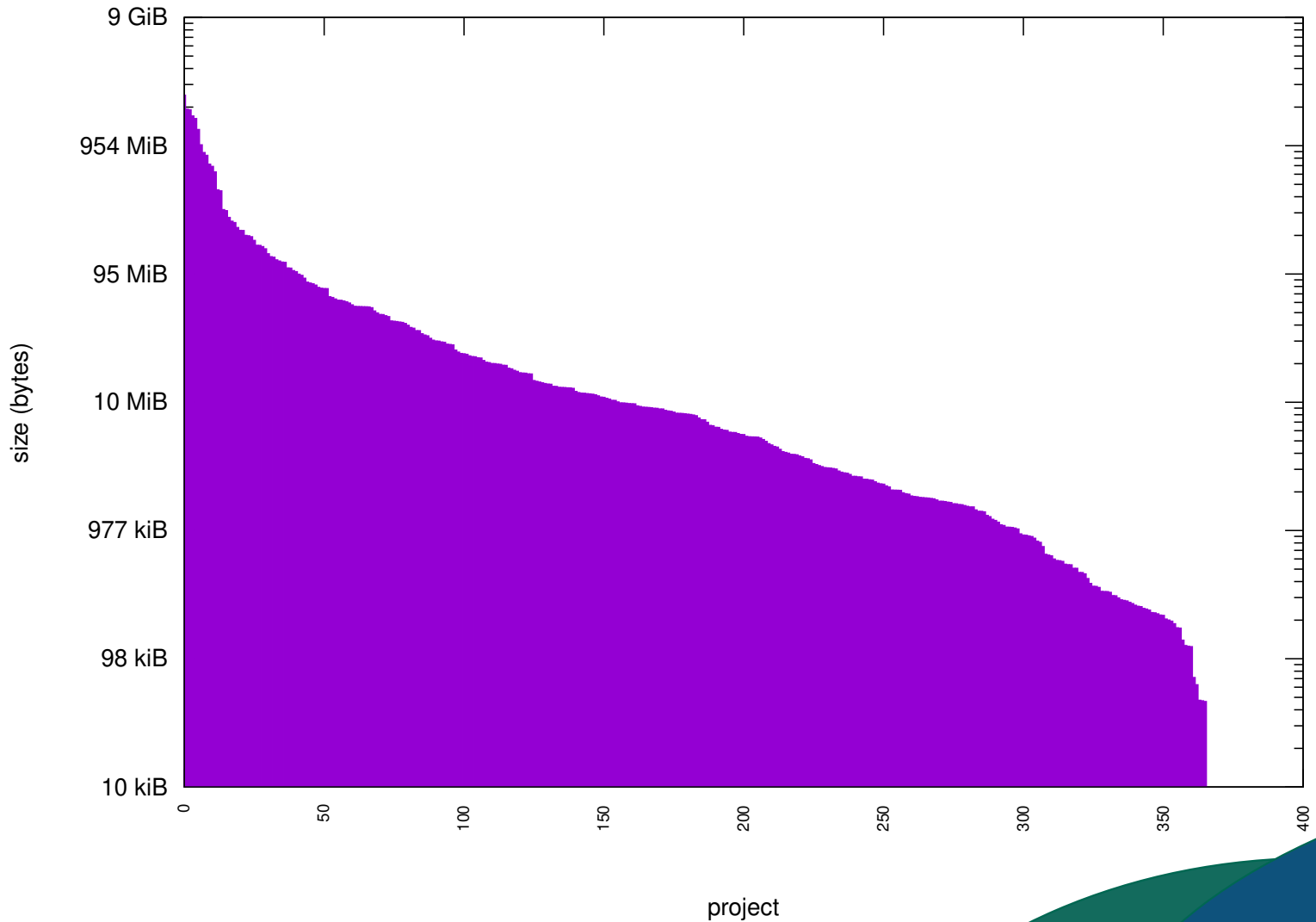
Number of symbols available per version



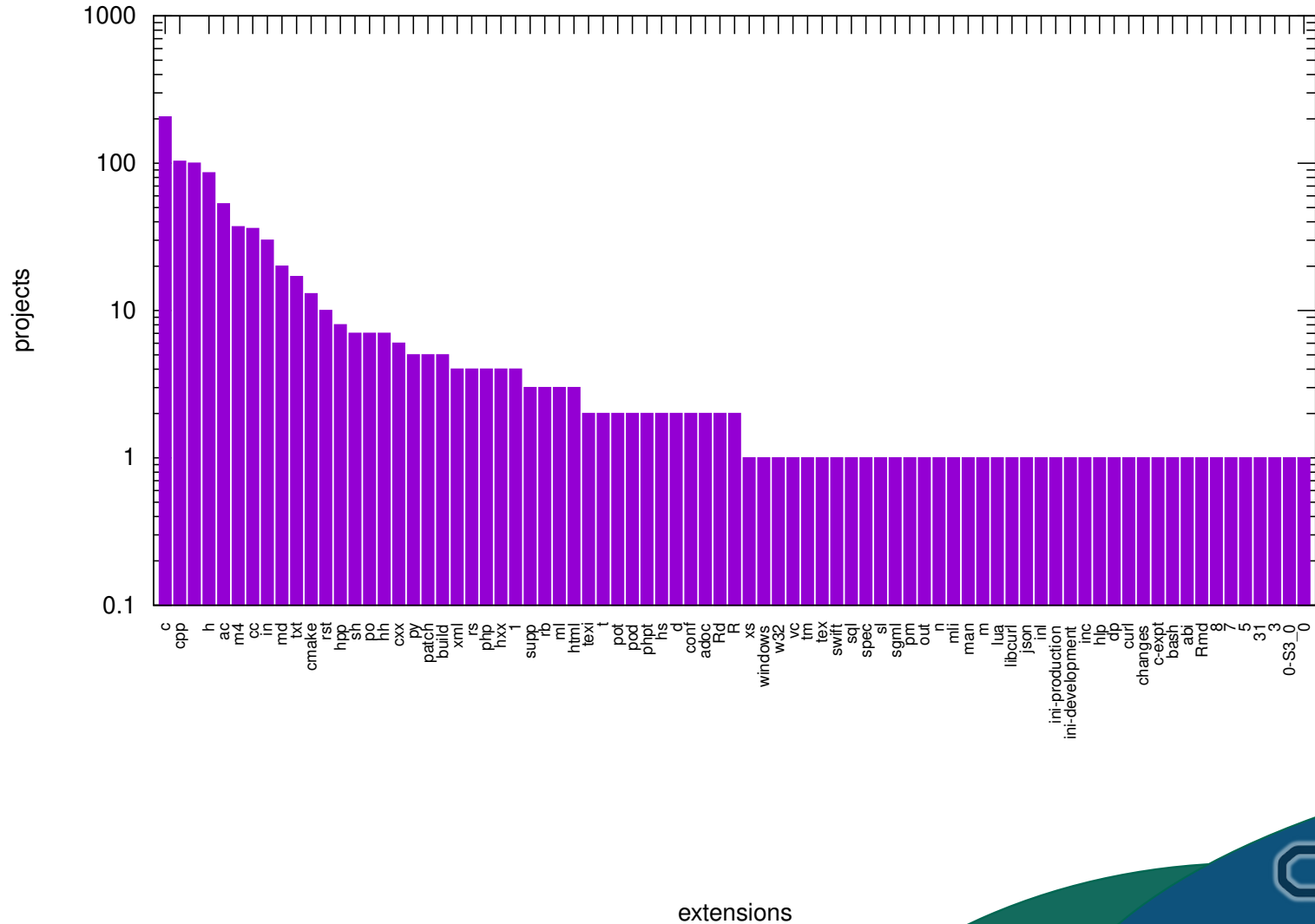
Number of files in analyzed projects



Size of analyzed projects

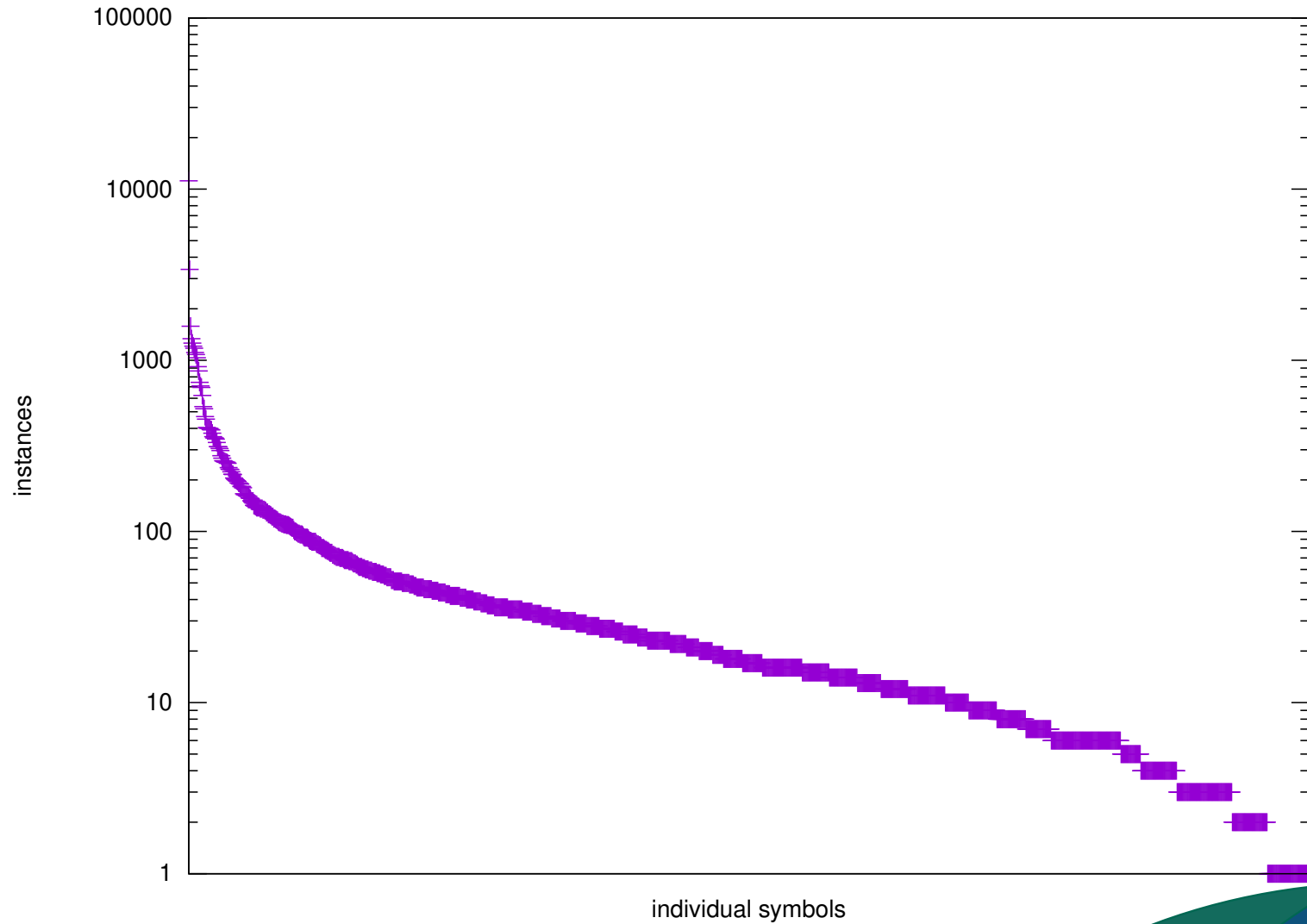


Projects using libcurl from each file extension

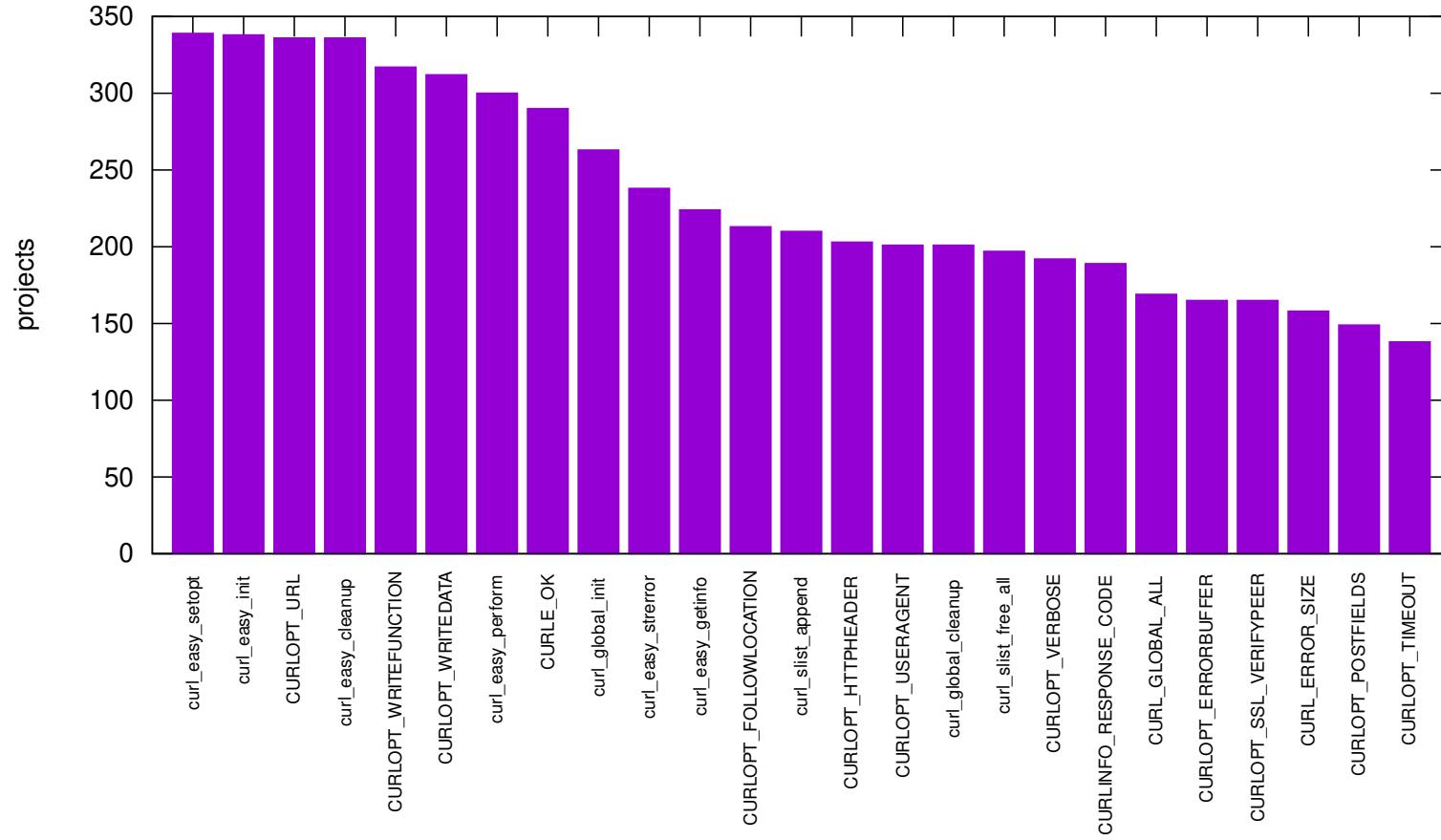


extensions

Total instances of each symbol across all codebases, ordered by popularity



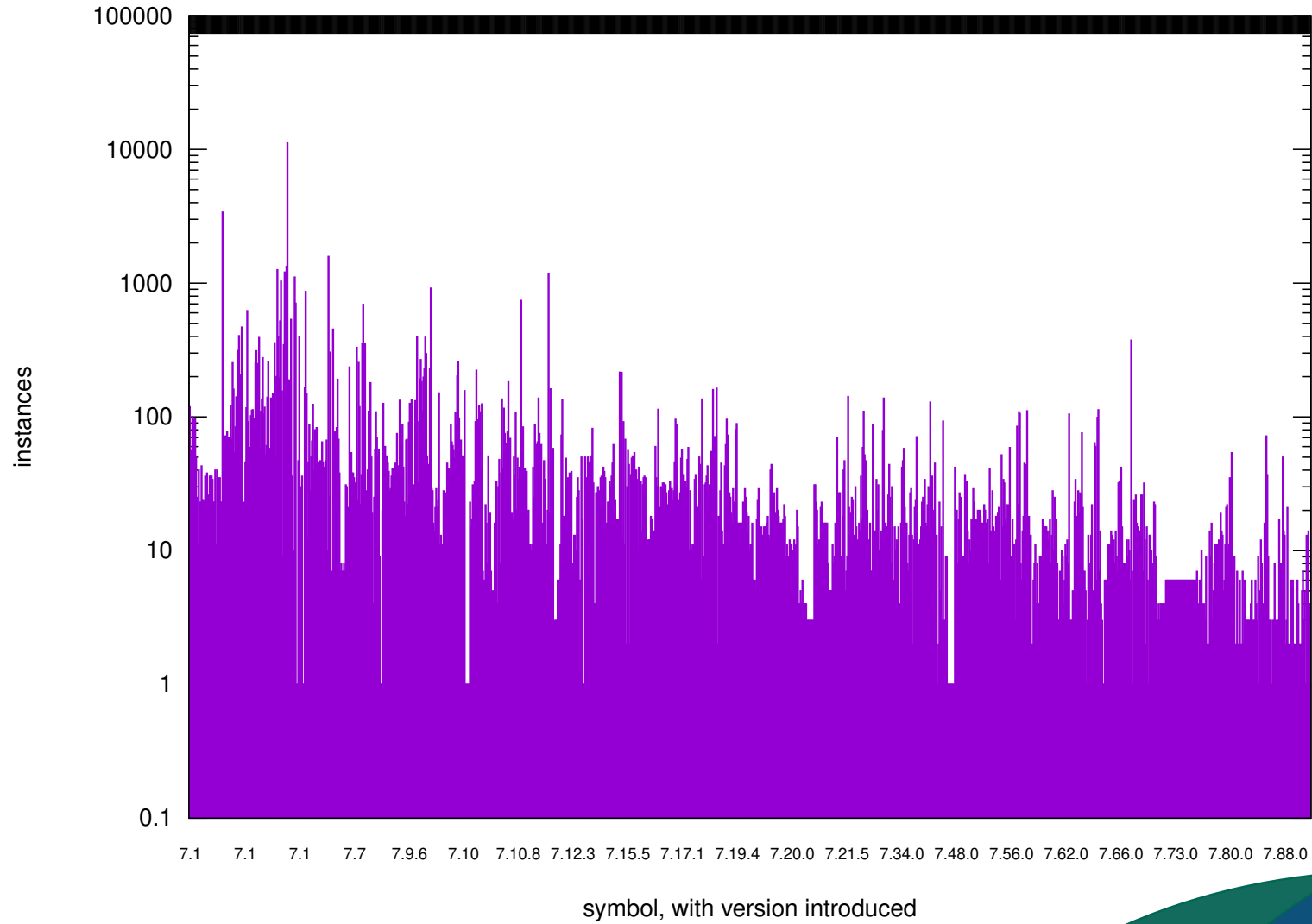
Symbols used by most projects



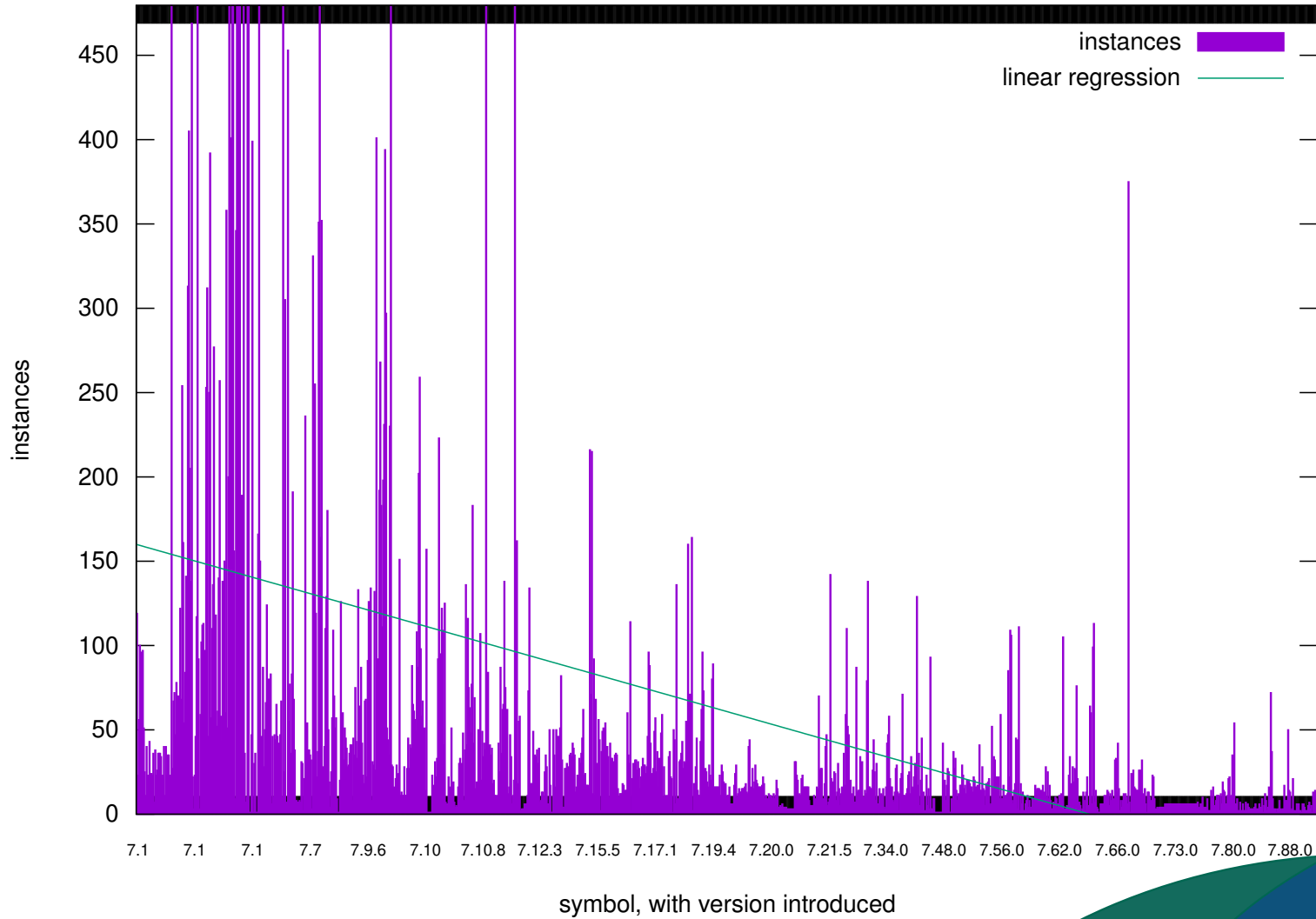
individual symbols



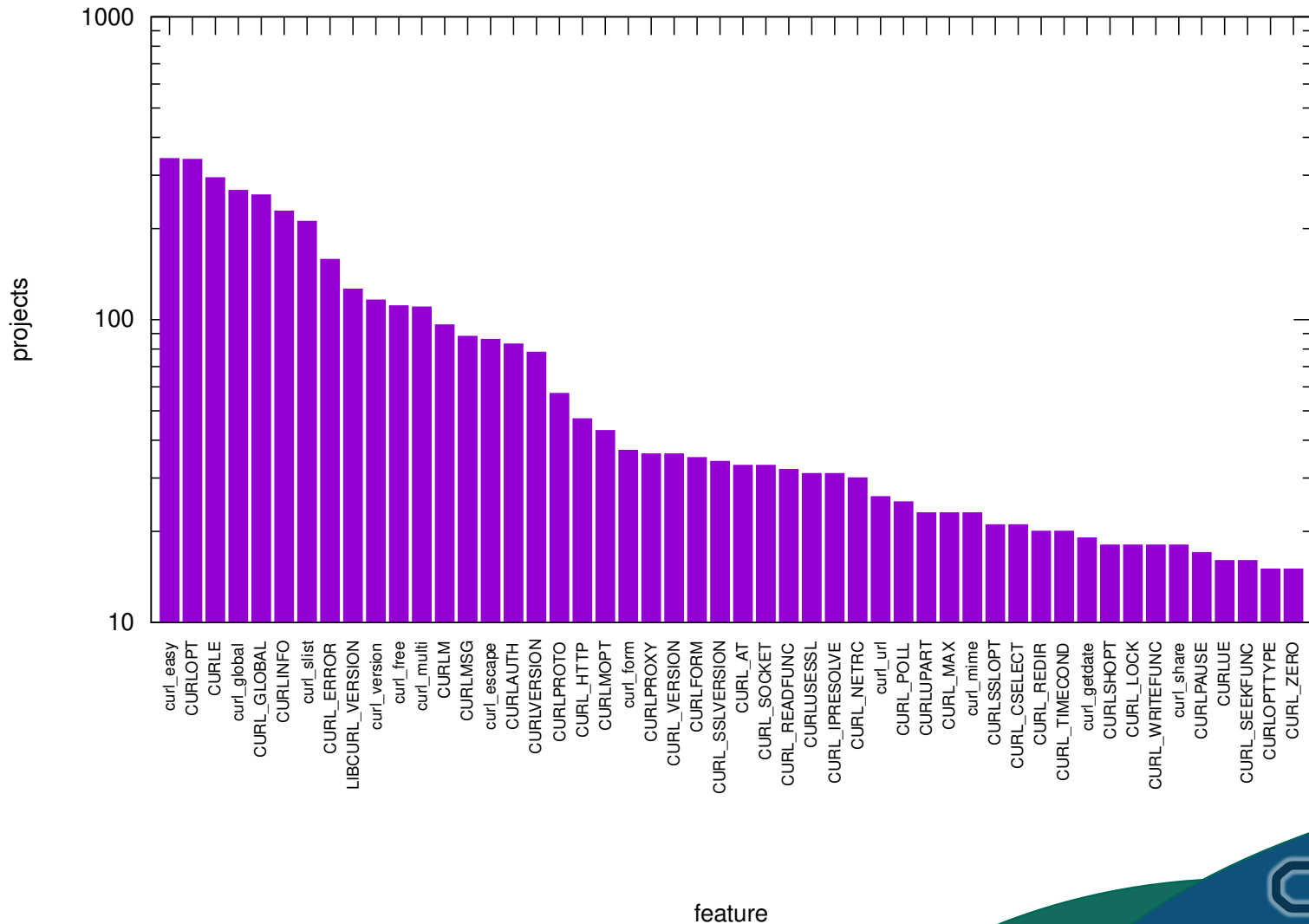
Instances of each symbol, ordered by version of introduction



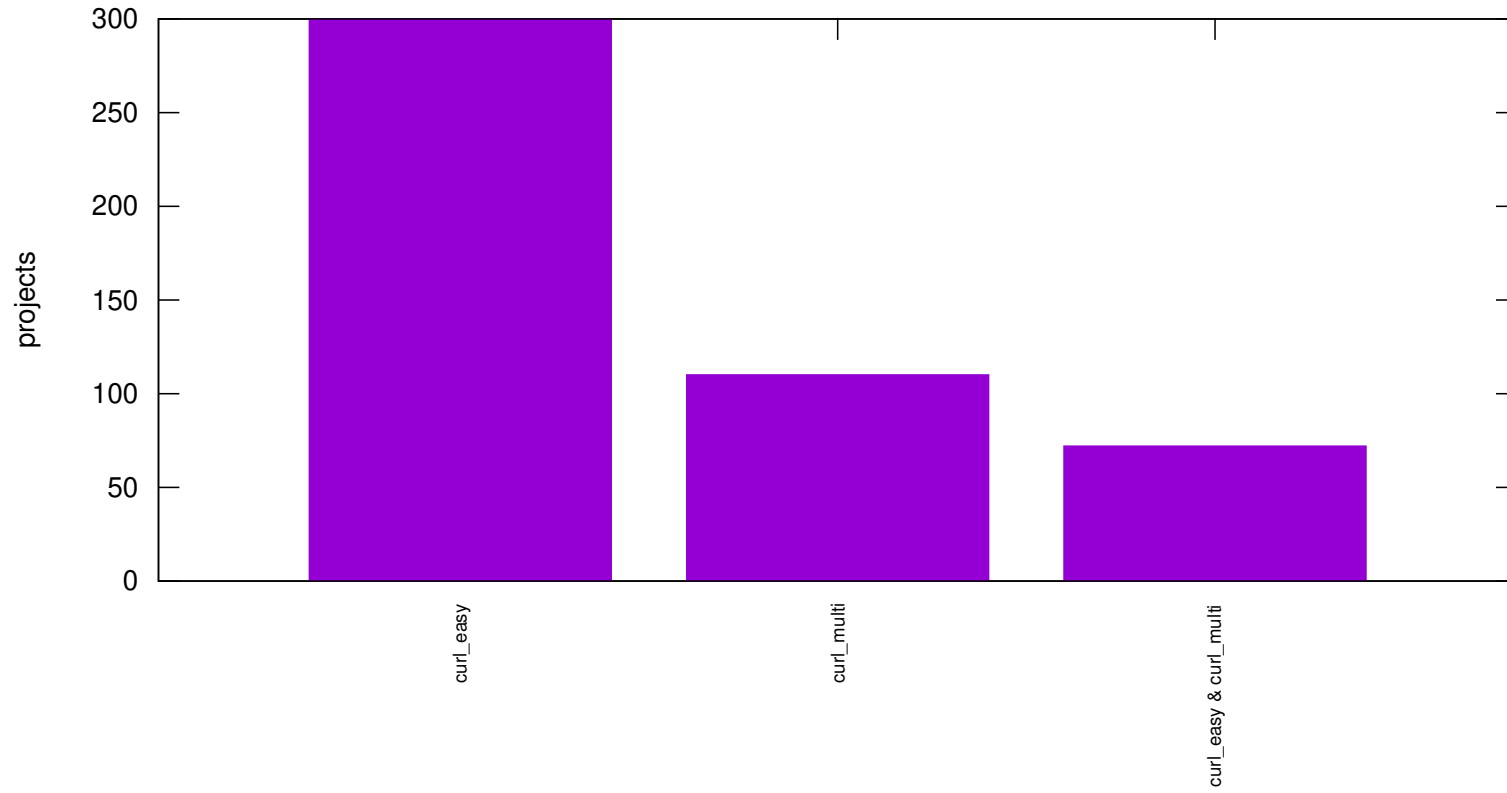
Instances of each symbol and trend, ordered by version of introduction



Projects using each category of libcurl features (top 50)



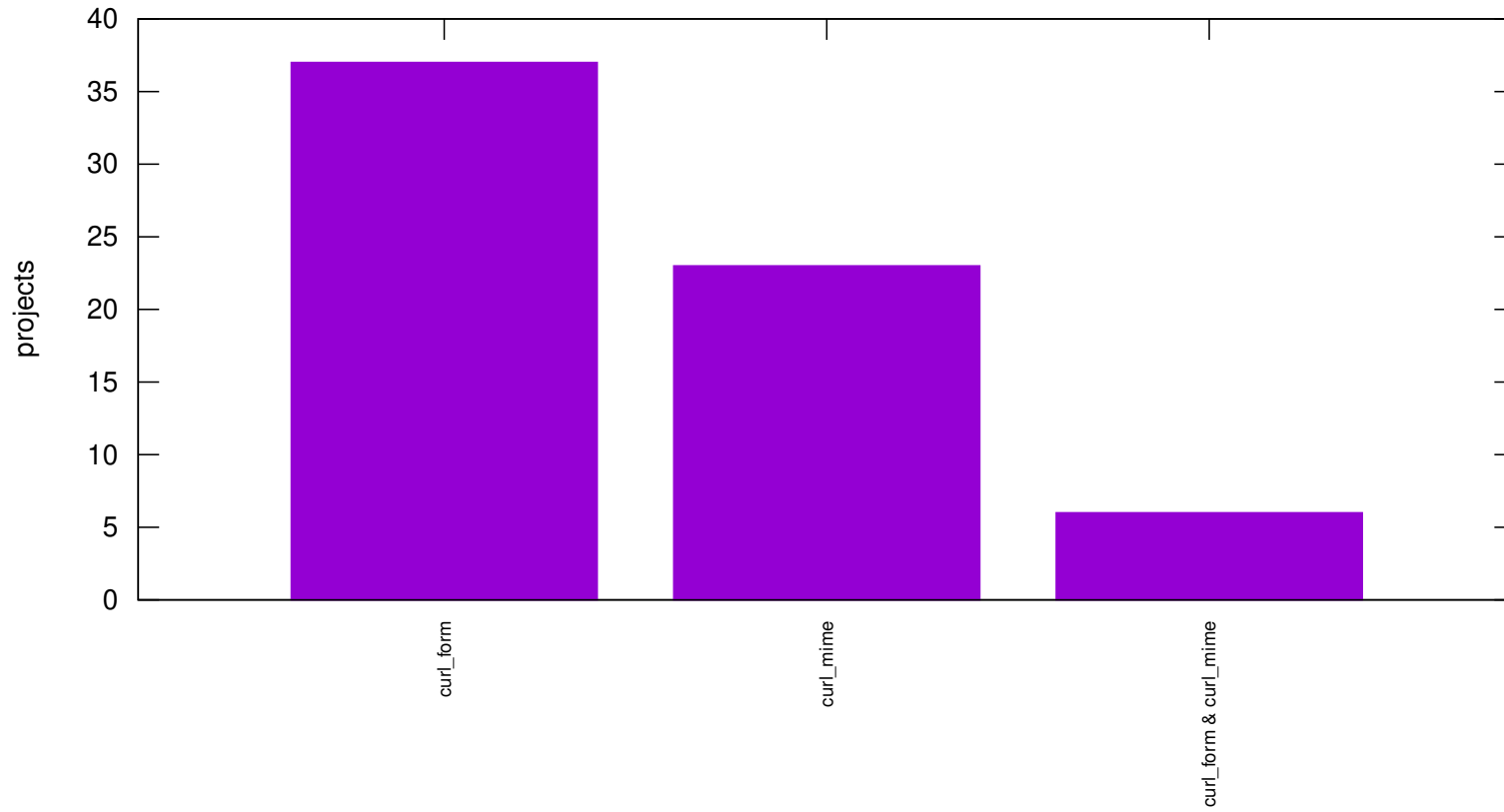
Projects using easy vs multi interface



interface



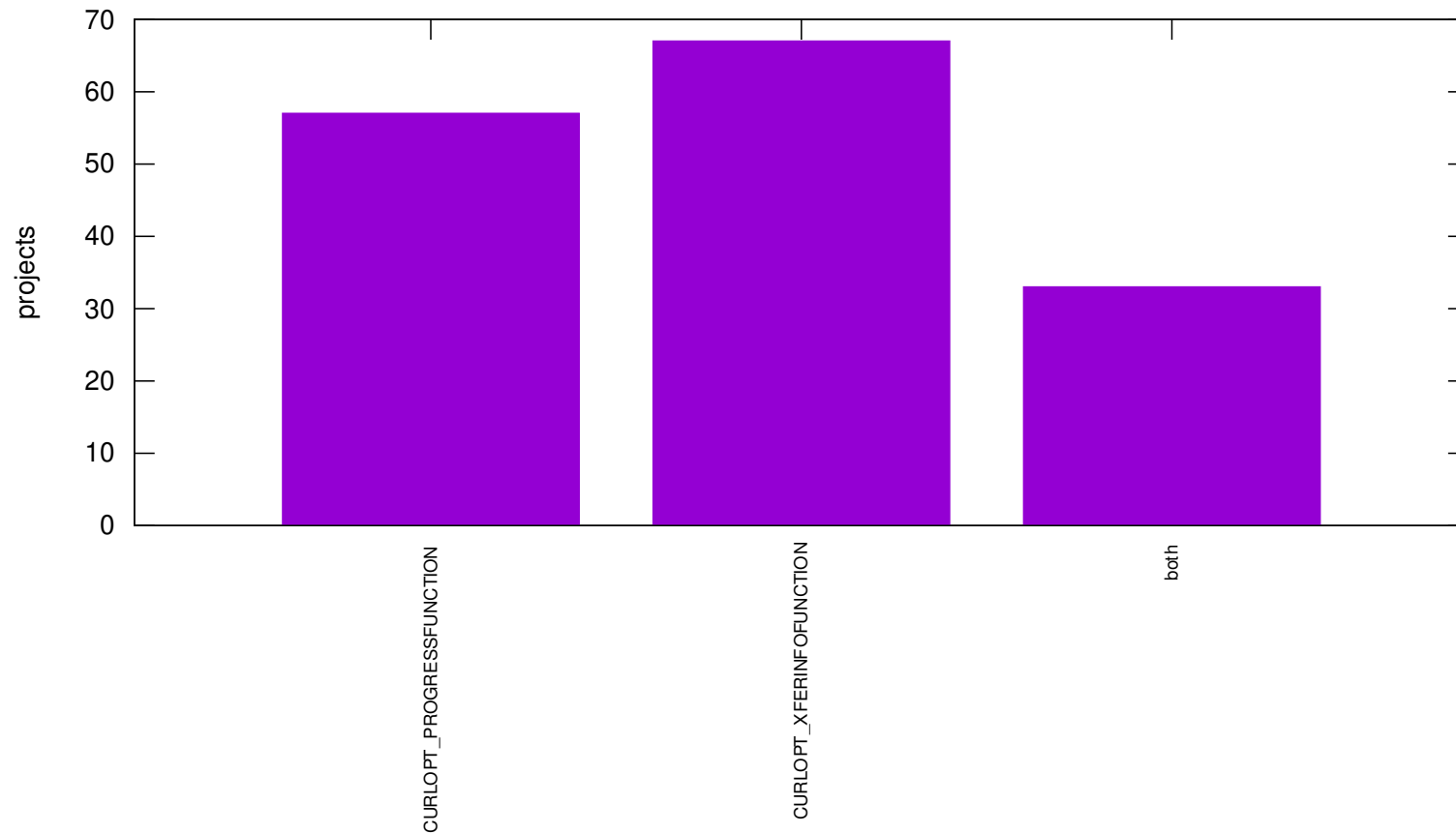
Projects using curl_form vs curl_mime interface



interface



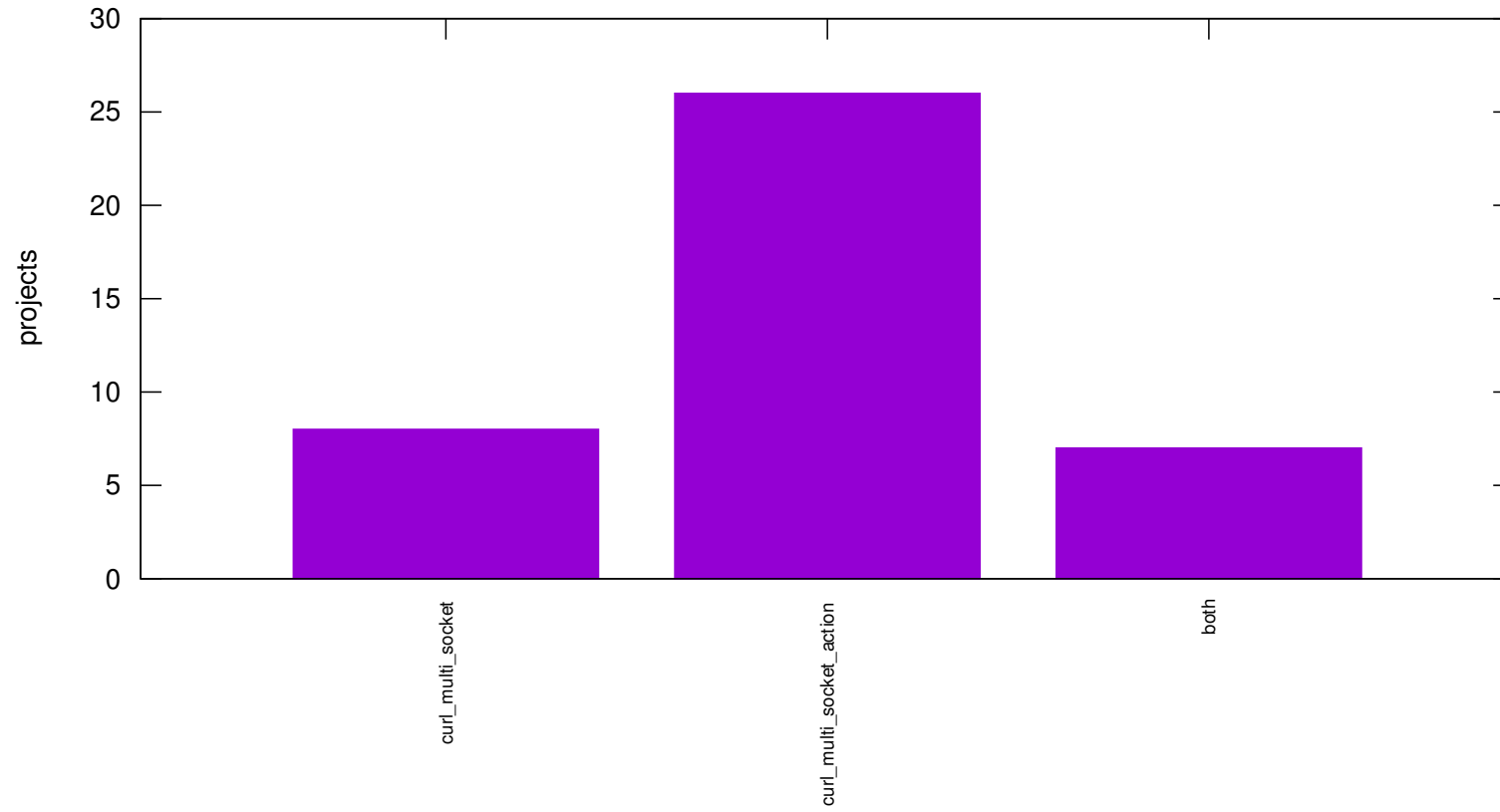
Projects using CURLOPT_PROGRESSFUNCTION vs CURLOPT_XFERINFOFUNCTION



interface



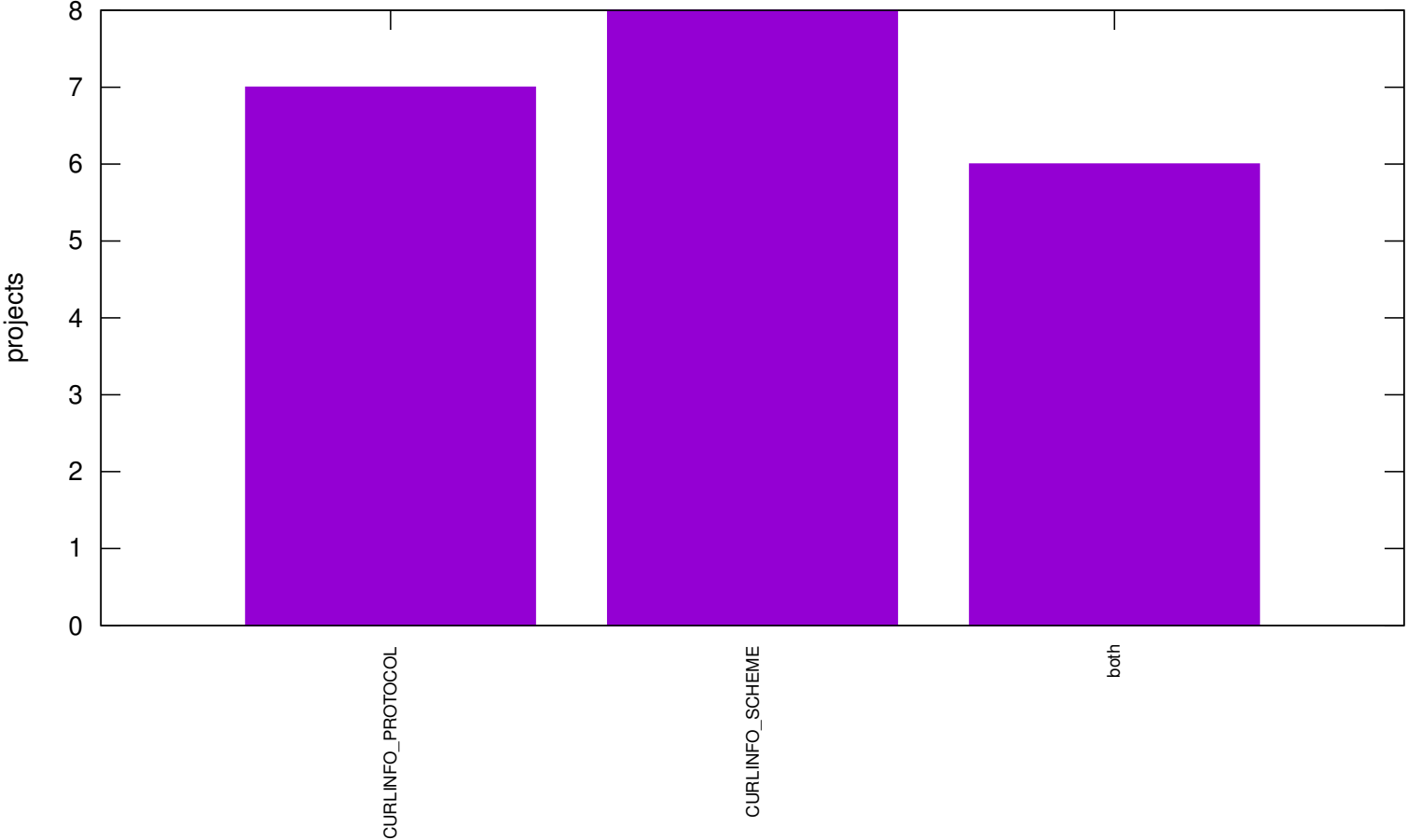
Projects using curl_multi_socket vs curl_multi_socket_action



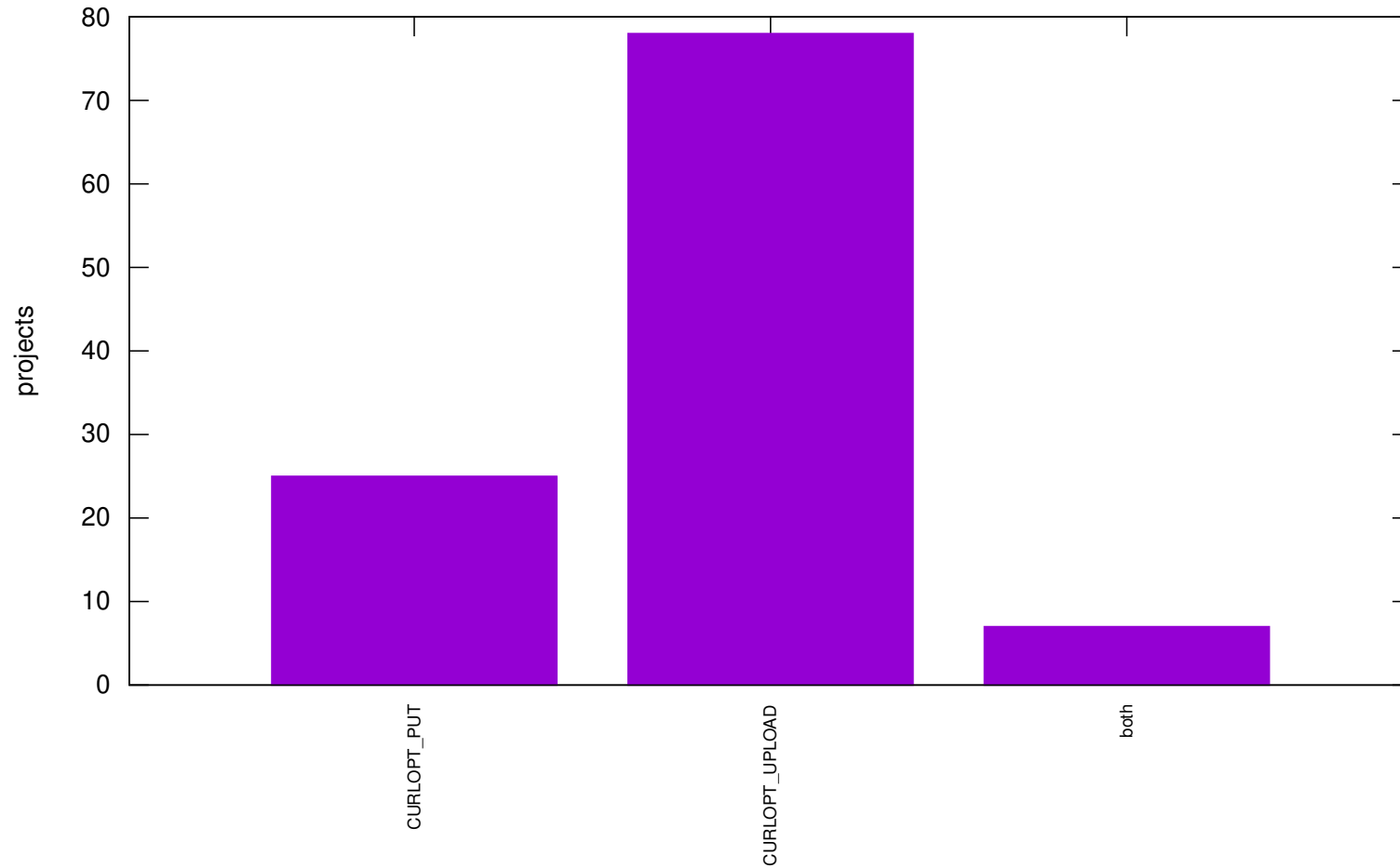
interface



Projects using CURLINFO_PROTOCOL vs CURLINFO_SCHEME



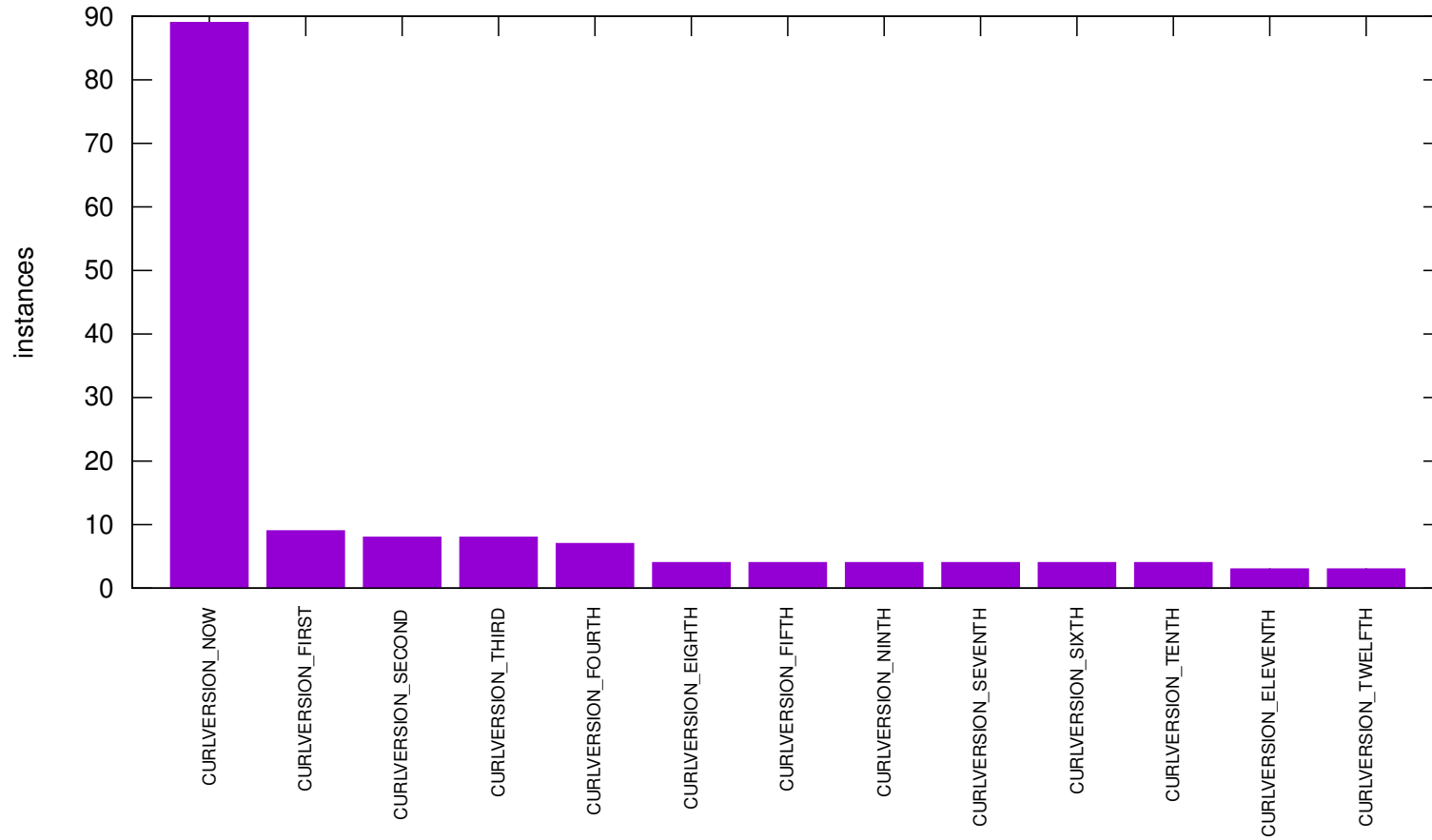
Projects using CURLOPT_PUT vs CURLOPT_UPLOAD



interface



CURLVERSION symbols by use



CURLVERSION symbols



Unused Symbols

CURLFOLLOW_ALL

CURLFOLLOW_FIRSTONLY

CURLFOLLOW_OBEYCODE

CURLINFO_HTTPAUTH_USED

CURLINFO_PROXYAUTH_USED

CURLMINFO_NONE

CURLMINFO_XFERS_ADDED

CURLMINFO_XFERS_CURRENT

CURLMINFO_XFERS_DONE

CURLMINFO_XFERS_PENDING

CURLMINFO_XFERS_RUNNING

CURLMNOTIFY_EASY_DONE

CURLMNOTIFY_INFO_READ

CURLMNBC_CLEAR_CONNS

CURLMNBC_CLEAR_DNS

CURLMNBC_NETWORK_CHANGED

CURLMNBC_NOTIFYDATA

CURLMNBC_NOTIFYFUNCTION

CURLMNBC_SSL_SIGNATURE_ALGORITHMS

CURLMNBC_UPLOAD_FLAGS

CURLULFLAG_ANSWERED

CURLULFLAG_DELETED

CURLULFLAG_DRAFT

CURLULFLAG_FLAGGED

CURLULFLAG_SEEN

CURLWS_NOAUTOPONG

CURL_HAS_DECLSPEC_ATTRIBUTE

curl_easy_ssls_export

curl_easy_ssls_import

curl_multi_get_offt

curl_multi_notify_disable

curl_multi_notify_enable

curl_ws_start_frame

Symbols Used Exactly Once*

CURLFORM_ARRAY_END	CURL_HTTPPOST_CALLBACK	CURL_PROGRESS_STATS	curl_multi_waitfds
CURLFORM_ARRAY_START	CURL_HTTPPOST_FILENAME	CURL_PUSH_ERROROUT	curl_mvaprintf
CURLOPT_NOTHING	CURL_HTTPPOST_LARGE	CURL_TRAILERFUNC_ABORT	curl_mvfprintf
CURLSSLBACKEND_AWSLC	CURL_HTTPPOST_PTRBUFFER	CURL_TRAILERFUNC_OK	curl_mvprintf
CURLSSLBACKEND_BORINGSSL	CURL_HTTPPOST_PTRCONTENTS	CURL_UPKEEP_INTERVAL_DEFAULT	curl_mvsprintf
CURLSSLBACKEND_LIBRESSL	CURL_HTTPPOST_PTRNAME	CURL_WIN32	curl_pushheader_byname
CURLSSLOPT_EARLYDATA	CURL_HTTPPOST_READFILE	LIBCURL_COPYRIGHT	curl_strequal
CURLU_DISALLOW_USER	CURL_LOCK_TYPE_CONNECT	LIBCURL_HAS	curl_strequal
CURLWS_OFFSET	CURL_LOCK_TYPE_COOKIE	curl_easy_nextheader	
CURL_EASY_NONE	CURL_LOCK_TYPE_DNS	curl_global_trace	
CURL_EASY_TIMEOUT	CURL_LOCK_TYPE_NONE	curl_mfprintf	
CURL_HET_DEFAULT	CURL_LOCK_TYPE_SSL_SESSION	curl_mprintf	
CURL_HTTPPOST_BUFFER	CURL_PROGRESS_BAR	curl_msprintf	

* But likely not *really* used



Future

Future

- What other data would be useful to have?

Future

- What other data would be useful to have?
- Can we use these data to inform future curl decisions?

Future

- What other data would be useful to have?
- Can we use these data to inform future curl decisions?
- What are we in the mood for deprecating now?

Thank-you!

Questions?