GNU Hackers Meeting 2020

GNU gnulib
Past, recent, and future work

Bruno Haible
<bruno@clisp.org>

GNU gnulib - Overview

- A source code library
- modular
- composed of:
  - POSIX and glibc compatible substitutes
  - General purpose code for applications (file system traversal, crypto, containers, i18n, etc.)
  - Maintainer infrastructure (config.guess, consistency checking tools, upload script for ftp.gnu.org, ...)

Containers

- Data structures with the same API across different implementations
  - different performance characteristics
  - code first, pick the implementation later
- Lists (array, circ. array, link, tree, hash)
- Set (array, hash, linkedhash) NEW (2018)
- Ordered set (array, tree)
- Map (array, hash, linkedhash) NEW (2018)
- Ordered map (array, tree) NEW (2018)

POSIX and glibc polyfills

- Header files and functions
- Lots of workarounds for platform bugs
- Usable in C
- Usable in C++ NEW
- Tested on
  - glibc systems, musl libc systems
  - macOS, FreeBSD, NetBSD, OpenBSD
  - AIX, HP-UX, IRIX, Solaris
  - Cygwin, mingw, MSVC
  - (Haiku, Minix, Android)

Multithreading APIs (1)

- The POSIX API NEW
  - modules pthread-thread, pthread-mutex, pthread-rwlock, pthread-once, pthread-cond, pthread-tss, sched_yield
- The ISO C 11 API NEW
  - modules thrd, mtx, cnd, tss
- The Gnulib API
  - modules thread, lock, cond, tls, yield
  - Implementation chosen at configure time:
    `--enable-threads={isoc,posix,windows}`
    NEW

Multithreading APIs (2)

- All 3 APIs are supported on all platforms!
  - except Minix
- Which API to use?
  - POSIX API is the best choice when you need extra POSIX functions (pthread_sigmask, pthread_atfork, ...)
  - Otherwise the Gnulib API is the best choice
  - ISO C API is not the best choice
    - e.g. lock initialization is clumsy

Multithreading (3)

- Document which functions are MT-safe under which conditions

```
_GL_MT_SAFE( IF_DIFFERENT(cd) )
size_t iconv (iconv_t cd,
              char **inbuf, size_t *inbytesleft,
              char **outbuf, size_t *outbytesleft);

_GL_MT_SAFE( IF_FIXED(program_name) )
void error (int status, int errnum, const char *message, ...);

_GL_MT_SAFE( IF_MT_SAFE(func) )
int pthread_once (pthread_once_t *once_control, void (*func) (void));

_GL_MT_SAFE( IF_NOT_CALLED(setlocale) )
char *nl_langinfo (nl_item item);
```

Beyond-BMP characters

- Unicode characters ≥ 0x10000
  - Emojis, math symbols, Chinese etc.
- Using GNU libunistring is one way to do it
  - But a lot of software still uses ISO C APIs
- `wchar_t` is 16-bit on Windows, 32-bit AIX
- `char32_t` is like wchar_t but always 32-bit
  - in ISO C 11, but hardly used yet
- Define `char32_t` based functions
  - character conversion, classification, width