Choosing System C library

Khem Raj
Comcast
Embedded Linux Conference Europe 2014 Düsseldorf Germany
Introduction

“God defined C standard library everything else is creation of man”
Introduction

• Standard library for C language
  • Provides primitives for OS service
  • Hosted/freestanding
  • String manipulations
  • Types
  • I/O
  • Memory
  • APIs
Linux Implementations

- GNU C library (glibc)
- uClibc
- eglibc
  - Now merged into glibc
- Dietlibc
- Klibc
- Musl
- bionic
Multiple C library FAQs

• Can I have multiple C libraries side by side?
• Can programs compiled with glibc run on uclibc or vice versa?
• Are they functional compatible?
• Do I need to choose one over other if I am doing real time Linux?
• I have a baremetal application what libc options do I have?
Posix Compliance

• Posix specifies more than ISO C
• Varying degree of compliance
What matters to you?

• Code Size
• Functionality
• Interoperability
• Licensing
• Backward Compatibility
• Variety of architecture support
• Dynamic Linking
• Build system
Codesize

- Dietlibc/klibc
  - Used in really small setup e.g. initramfs
- Bionic
  - Small linked into every process
- uClibc
  - Configurable
    - Size can be really small at the expense of functionality
- Eglibc
  - Has option groups can be ( < 1M )
License

- Bionic – BSD/Apache-2.0
- Musl – MIT
- Uclibc – LGPL-2.1
- Eglibc/Glibc – LGPL-2.1 Assigned to FSF
- Dietlibc – GPLv2
- Klibc – GPLv2
- Newlib – some parts are GPLv3
Compliance

• Musl strives for ISO/C and POSIX compliance
No-mmu

• uClibc supported No-mmu
Distributions

• Glibc is used in majority of distribution
  – Fedora-like, RHEL, Debian-like, SuSE, Gentoo, Archlinux ....

• Buildroot
  – Uclibc and glibc

• OpenEmbedded
  – Uclibc/glibc/musl

• Alpine
  – Uclibc/musl
Performance

• Glibc has architecture optimized str/mem routines
Stability

- Stable ABI – glibc, musl
- Backward compatibility – glibc, musl
- Symbol versioning – glibc
- LSB compliance - glibc
Features

- C99 Math library – glibc and musl
- Posix threads – uclibc/glibc/musl
- C11 TLS – uclibc/glibc/musl
- Wide chars – uclibc/glibc/musl
- Profiling – glibc
- Debugging Features – glibc
- Zoneinfo – glibc/musl
Features

• Stack smashing Protection – uclibc/glibc/musl
• Heap corruption Detection – musl/glibc
CPU Architectures

• Glibc supports the widest range
  – ARM/MIPS/x86/SH/PPC/SPARC/Alpha/IA64/Microblaze/s390

• Uclibc does support most of above except few. Eg. S390, x32 but then it supports ARC, AVR32..

• Musl is getting architecture parity

• Bionic arm/x86/mips