Code Analysis of uClibc and Klee-uClibc

Code Structure of Libc
One folder per library.
One folder for the dependencies for each processor architecture.

Libraries
unistd
termios
string
stdlib
stdio
signal
inet
pwd_group
misc

There are platform specific hiding of features as illustrated below:

```
#if defined __USE_BSD || (defined __USE_XOPEN && !defined __USE_UNIX98)

libc_hidden_proto(fstat)
libc_hidden_proto(fopen)
libc_hidden_proto(fclose)
libc_hidden_proto(__fsetlocking)
libc_hidden_proto(fileno)
libc_hidden_proto(fgets_unlocked)
#ifdef __UCLIBC_HAS_XLOCALE__
libc_hidden_proto(__ctype_b_loc)
#elif __UCLIBC_HAS_CTYPE_TABLES__
libc_hidden_proto(__ctype_b)
#endif

#endif
```

These are done to ensure that the features from the standard library are being over ridden by the corresponding methods in Klee-uClibc implementation.

libc_hidden_proto(method) is used to disable the defintion of the method in the Standard C library. The reason for disabling the definitions of some of the methods in the Klee-uClibc needs to be investigated though, especially the way the methods to be disabled are selected.

In general, the differences between Klee-uClibc and uClibc are the over riding of certain methods in the uClibc library with the implementations in Klee-uClibc.

The syntax of a method that over rides the Standard C library method is as shown below:

```
extern void *__exec_alloc(size_t size, int func) attribute_hidden;
void attribute_hidden *__exec_alloc(size_t size, int func)
{
        function_body
}
```

The *attribute_hidden* is a macro used to hide the implementation of the standard library.

Some files are missing #define _GNU_SOURCE. This is done to disable the compiling of the GNU

extensions.

System Dependencies

The system dependecies contain files speciffic to different platforms such as x86, sparc, etc. They also contain the register information and assembly files. One interesting aspect about this folder is that the Klee-uClibc adds support for the 64 bit processors which is not found in uClibc.
Klee-uClibc has made a few changes to some of the architectures such as adding architecture specific features for system calls like abort() (e.g. sysdeps/linux/v850/bits/uClibc_arch_features.h).

System Calls

There is no difference in terms of the system calls used internally in the macros. But, in Klee-uClibc, it is implemented as a strong alias as against weak alias in uClibc.

References

1. http://www.scs.stanford.edu/histar/src/pkg/uclibc/include/libc-symbols.h
2. http://www.bottomupcs.com/libraries_and_the_linker.html