iBCS2 Emulation Under Linux.

Overview

- Explaination of iBCS2 API and SVr4 API
- Kernel internals
- Native compilation of SCO/SVr4 programs.
- Installation.
- Shared libraries
- Troubleshooting/tracing
- Limitations
- Applications known to work.

Introduction

iBCS2 emulator allows people to run binaries from other i386 Unix OS's on their Linux machines. Supported systems include

- SVR4 (Solaris, Interactive, Unixware, USL, Dell etc.)
- SVR3 generic
- SCO (SVR3 with extensions for symlinks and long filenames)
- Wyse V/386 (SVR3 with extensions for symlinks)
- Xenix V/386 (386 small model binaries only)
- Xenix 286
- i386 BSD (386BSD, FreeBSD, NetBSD, BSDI/386) - very alpha.

Shared libraries are provided to help run binaries for SCO and SVr4 variants.

Bootstrapping the market.

- Mispercetion that people will not spend money to purchase software for a free OS.
- Size of user base not easy to determine.
- Percieved by some as a “hacker’s OS”.

The iBCS2 package allows us to:

- Demonstrate that linux community wants commercial applications.
- Show that people will spend money, if the product is good.
- Gives leverage for a native port to linux.

Progress since last year.

- New binutils can be used to compile/link COFF/ELF images, shared libs.
- `mprotect` now works under linux.
- IPC works better.
- Keymaps now much better - uses loadable keymaps.
- Some support for running BSD binaries now available (very ALPHA, not iBCS2).

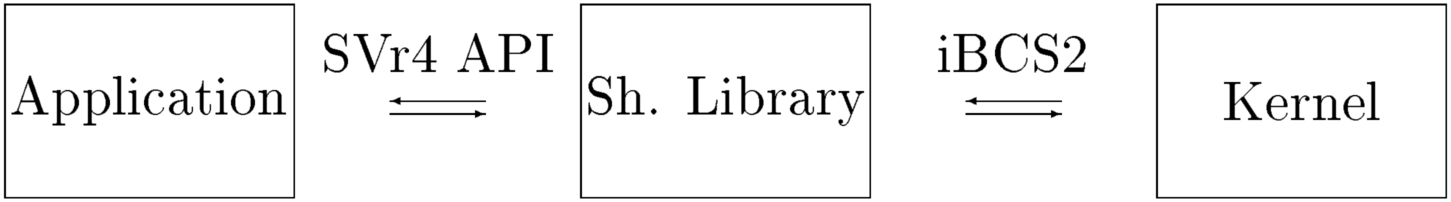
Explaination of iBCS2 Standard.

- Provides binary compatibility between different implementations of Unix.
- Reference implementation is SVr3.
- Specification is for application ↔ kernel interface.
- SCO is based upon SVr3, but includes some extensions.
- Object/Executable file format is COFF.
- ISC and Wyse are other implementations based upon SVr3.

Even though this level of the standard is dated, if an application is ported to any Unix variant, it will probably be first ported to SCO.

Explaination of SVr4 API

- Backwards compatible with SVr3. SCO binaries also run.
- Extensions to iBCS2 are not always compatible with SCO extensions to iBCS2 - emulator must have personality dependent mappings.
- Object/Executable file format is ELF.
- Shared libraries use runtime dynamic linking - elegant and very easy to generate and use.
- SVr4 ABI specification is for application ↔ shared library level:



Linux kernel internals for iBCS2.

- Program loaders for COFF, ELF and x.out formats.
- Kernel “`lcall $7,0`” trap.
- Translate syscall numbers - personality dependent.
- Translate error, signal numbers.
- Translate arguments, structures as required for some syscalls.

Native Compilation of COFF iBCS2 binaries.

- New binutils contain support for ELF, COFF and a.out
- GCC can easily be configured to generate COFF assembly files.
- Runtime shared library seems to work for some binaries.
- SVr3/SCO header files not available.
- Linking `libc.a` planned, but not ready.
- If your licence permits it, you can copy libraries and headers from SCO system.

Native Compilation of ELF iBCS2 binaries.

- Native ELF tools can be used to compile and link SVr4 applications.
- SVr4 compatible header files not available - subset of files in libc_s source tree, but not really suitable for applications.
- Runtime shared library/dynamic linker works well for many applications.
- Binutils 2.N can be configured to simultaneously support multiple file formats.
- Under Solaris, Sun CC generates debugging stabs which gdb cannot understand. Use gcc if you want to be able to debug something under Linux.
- If your licence permits it, you can copy libraries and headers from SVr4 system. Do not bother with `libnsl.so` from a SVr4 system - it will not work.

Installation of emulator.

- Current version requires 1.2 or latest 1.3.N kernel once 1.3 starts.
- If your system came from a distribution, use the version of the emulator that came with the distribution.
- No kernel patches required.
- iBCS2 emulator designed to work as a loadable module.
- Can be unloaded to save memory if not being used.

COFF Shared libraries

- Version of `libc_s` available - works with some binaries.
- No linux version of `libnsl_s` available.
- iBCS2 specifies `libX11_s` without jump table - this portion of standard ignored. XFree86 shared libraries available, but not compatible with SCO X libraries.
- Many early commercial versions of SVr3/SCO were buggy - most SVr3/SCO binaries are now static linked.

ELF Shared libraries

- Version of `libc.so` available - works with many applications.
- Dynamic linker included in `libc.so`. Separate version of `ld.so` also available for Solaris binaries.
- TLI based networking not supported - socket based networking should work.
- SVr4/ELF shared libraries should be put in /usr/i486-sysv4/lib instead of /usr/lib. Symbolic links are required:

```
/usr/lib/libc.so.1 -> /usr/i486-sysv4/lib/libc.so.1
/usr/lib/ld.so.1   -> /usr/i486-sysv4/lib/ld.so.1
```

- UnixWare X libraries directly call STREAMS/TLI instead of sockets. UnixWare X binaries unsuitable to be used under Linux.

Troubleshooting/Tracing

- Tracing similar to `truss` available - can be used to debug emulator and applications.
- Some applications do not install cleanly - list of known problems is shipped with emulator source code.
- Precompiled libraries are stripped of debugging information - you can build the libraries yourself to debug library problems.

gdb

Get the file pub/linux/packages/GCC/gdb-4.14.bin.tar.gz Has the following capability:

```
(gdb) break main
Breakpoint 1 at 0x80004b3: file hello.c, line 5.
(gdb) run
Starting program: /usr/users/eric/hello

Breakpoint 1, main () at hello.c:5
(gdb) info shared
From        To          Syms Read   Shared Object Library
0x50006000  0x500998bc  Yes         /lib/elf/libc.so.4
0x50000000  0x50004954  Yes         /lib/elf/ld-linux.so.1
(gdb) break printf
Breakpoint 2 at 0x50035d90
(gdb) cont
Continuing.

Breakpoint 2, 0x50035d90 in _IO_printf ()
(gdb)
```

Limitations

- Applications that require STREAMS, TLI networking will fail. Sockets OK.
- SVr4 shared libraries may be missing entry points.
- `/proc` filesystem not emulated.

How to help the project.

SVr4 libraries need more testing and work. Binutils for native ELF can be used to compile libraries with debugging.

Easiest to get the libc.so.1 source tree and compile with debugging. If application was built with debugging, you can debug both application/library as if it were one large application.

Trace tells you which syscalls were made, and what the results were.

To add a missing API function:

- libc-linux may already have suitable sources for the missing function.
- Modify Makefile to add new file to source list.
- Rebuild library - function is automatically incorporated.
- Send me the patches.

Support for STREAMS would be nice at some point - kernel itself needs to support this first.

Applications known to work.

Partial list:

- WordPerfect - both character based and X based.
- Oracle - database manager.
- Informix - relational database management system.
- Xess - X based spreadsheet (SCO app - built under Linux).
- CorelDraw - Drawing/paint program.
- ZMail - GUI based mail program.

Copyright issues.

Information for development of emulator and libraries obtained from:

- Books on iBCS2, SVr4 API, etc.
- man pages.
- Manuals.
- Header files.
- Behavoir of existing programs under Linux.
- Drafts of ABI+ standard.

None of the developers has ever had access to any controlled Unix sources.

Contributors.

- Al Longyear - COFF loader, emulator, libc_s.
- Mike Jagdis - x.out loader, emulator, Wyse and BSD support.
- Joe Portman - emulator development, libc_s.
- Phillip Balister - emulator, COFF gdb.
- Drew Sullivan - Source tree maintainer.
- Karl Kiniger - Interactive 4.0 support.
- Eric Youngdale - ELF loader, v0.1 of emulator, libc_s.

Conclusions.

- Native versions of applications will always be preferred - lower memory comsumption. Support more likely with native version.
- Success of SCO/SVr4 version under linux may convince vendors to do a native port to better support the users.
- Many commercial applications will never be ported, for various reasons - running SCO/SVr4 binaries under Linux works fairly well.