NAME

        perltru64 - Perl version 5 on Tru64 (formerly known as Digital UNIX formerly known as DEC OSF/1) systems

DESCRIPTION

        This document describes various features of HP's (formerly Compaq's, formerly Digital's) Unix operating system (Tru64) that will affect how Perl version 5 (hereafter just Perl) is configured, compiled and/or runs.

      Compiling Perl 5 on Tru64

        The recommended compiler to use in Tru64 is the native C compiler. The native compiler produces much faster code (the speed difference is noticeable: several dozen percentages) and also more correct code: if you are considering using the GNU C compiler you should use at the very least the release of 2.95.3 since all older gcc releases are known to produce broken code when compiling Perl. One manifestation of this brokenness is the lib/sdbm test dumping core; another is many of the op/regexp and op/pat, or ext/Storable tests dumping core (the exact pattern of failures depending on the GCC release and optimization flags).

        Both the native cc and gcc seem to consume lots of memory when building Perl. toke.c is a known trouble spot when optimizing: 256 megabytes of data section seems to be enough. Another known trouble spot is the mktables script which builds the Unicode support tables. The default setting of the process data section in Tru64 should be one gigabyte, but some sites/setups might have lowered that. The configuration process of Perl checks for too low process limits, and lowers the optimization for the toke.c if necessary, and also gives advice on how to raise the process limits (for example: `ulimit -d 262144`)

        Also, Configure might abort with

```
Build a threading Perl? [n]
Configure[2437]: Syntax error at line 1 : 'config.sh' is not expected.
```

        This indicates that Configure is being run with a broken Korn shell (even though you think you are using a Bourne shell by using "sh Configure" or "./Configure"). The Korn shell bug has been reported to Compaq as of February 1999 but in the meanwhile, the reason ksh is being used is that you have the environment variable BIN_SH set to 'xpg4'. This causes /bin/sh to delegate its duties to /bin/posix/sh (a ksh). Unset the environment variable and rerun Configure.

      Using Large Files with Perl on Tru64

        In Tru64 Perl is automatically able to use large files, that is, files larger than 2 gigabytes, there is no need to use the Configure -Duselargefiles option as described in INSTALL (though using the option is harmless).

      Threaded Perl on Tru64

        If you want to use threads, you should primarily use the Perl 5.8.0 threads model by running Configure with -Duseithreads.

        Perl threading is going to work only in Tru64 4.0 and newer releases, older operating releases like 3.2 aren't probably going to work properly with threads.

        In Tru64 V5 (at least V5.1A, V5.1B) you cannot build threaded Perl with gcc because the system header <pthread.h> explicitly checks for supported C compilers, gcc (at least 3.2.2) not being one of them. But the system C compiler should work just fine.

      Long Doubles on Tru64

        You cannot Configure Perl to use long doubles unless you have at least Tru64 V5.0, the long double support simply wasn't functional enough before that. Perl's Configure will override attempts to use the long doubles (you can notice this by Configure finding out that the *modfl()* function does not work as it should).

        At the time of this writing (June 2002), there is a known bug in the Tru64 libc printing of long doubles when not using "e" notation. The values are correct and usable, but you only get a limited number of digits displayed unless you force the issue by using `printf "%.33e",$num` or the like. For Tru64 versions V5.0A through V5.1A, a patch is expected sometime after perl 5.8.0 is released. If your libc has not yet been patched, you'll get a warning from Configure when selecting long doubles.

DB_File tests failing on Tru64

The DB_File tests (db-btree.t, db-hash.t, db-recno.t) may fail you have installed a newer version of Berkeley DB into the system and the -I and -L compiler and linker flags introduce version conflicts with the DB 1.85 headers and libraries that came with the Tru64. For example, mixing a DB v2 library with the DB v1 headers is a bad idea. Watch out for Configure options -Dlocincpth and -Dloclibpth, and check your /usr/local/include and /usr/local/lib since they are included by default.

The second option is to explicitly instruct Configure to detect the newer Berkeley DB installation, by supplying the right directories with `-Dlocincpth=/some/include` and `-Dloclibpth=/some/lib` **and** before running "make test" setting your LD_LIBRARY_PATH to */some/lib*.

The third option is to work around the problem by disabling the DB_File completely when build Perl by specifying -Ui_db to Configure, and then using the BerkeleyDB module from CPAN instead of DB_File. The BerkeleyDB works with Berkeley DB versions 2.* or greater.

The Berkeley DB 4.1.25 has been tested with Tru64 V5.1A and found to work. The latest Berkeley DB can be found from <http://www.sleepycat.com>.

64−bit Perl on Tru64

In Tru64 Perl's integers are automatically 64-bit wide, there is no need to use the Configure -Duse64bitint option as described in INSTALL. Similarly, there is no need for -Duse64bitall since pointers are automatically 64-bit wide.

Warnings about floating-point overflow when compiling Perl on Tru64

When compiling Perl in Tru64 you may (depending on the compiler release) see two warnings like this

```
cc: Warning: numeric.c, line 104: In this statement, floating-point
overflow occurs in evaluating the expression "1.8e308". (floatoverfl)
return HUGE_VAL;
-----------^
```

and when compiling the POSIX extension

```
cc: Warning: const-c.inc, line 2007: In this statement, floating-point
overflow occurs in evaluating the expression "1.8e308". (floatoverfl)
return HUGE_VAL;
-------------------^
```

The exact line numbers may vary between Perl releases. The warnings are benign and can be ignored: in later C compiler releases the warnings should be gone.

When the file *pp_sys.c* is being compiled you may (depending on the operating system release) see an additional compiler flag being used: `-DNO_EFF_ONLY_OK`. This is normal and refers to a feature that is relevant only if you use the `filetest` pragma. In older releases of the operating system the feature was broken and the NO_EFF_ONLY_OK instructs Perl not to use the feature.

Testing Perl on Tru64

During "make test" the `comp/cpp` will be skipped because on Tru64 it cannot be tested before Perl has been installed. The test refers to the use of the `-P` option of Perl.

ext/ODBM_File/odbm Test Failing With Static Builds

The ext/ODBM_File/odbm is known to fail with static builds (Configure -Uusedl) due to a known bug in Tru64's static libdbm library. The good news is that you very probably don't need to ever use the ODBM_File extension since more advanced NDBM_File works fine, not to mention the even more advanced DB_File.

Perl Fails Because Of Unresolved Symbol sockatmark

If you get an error like

```
 Can't load '.../OSF1/lib/perl5/5.8.0/alpha-dec_osf/auto/IO/IO.so' for module IO:
```

you need to either recompile your Perl in Tru64 4.0D or upgrade your Tru64 4.0D to at least 4.0F: the

sockatmark() system call was added in Tru64 4.0F, and the IO extension refers that symbol.

read_cur_obj_info: bad file magic number

You may be mixing the Tru64 cc/ar/ld with the GNU gcc/ar/ld. That may work, but sometimes it doesn't (your gcc or GNU utils may have been compiled for an incompatible OS release).

Try 'which ld' and 'which ld' (or try 'ar --version' and 'ld --version', which work only for the GNU tools, and will announce themselves to be such), and adjust your PATH so that you are consistently using either the native tools or the GNU tools. After fixing your PATH, you should do 'make distclean' and start all the way from running the Configure since you may have quite a confused situation.

AUTHOR

Jarkko Hietaniemi <jhi@iki.fi>