NCURSES ANNOUNCE

Announcing ncurses 5.4

The ncurses (new curses) library is a free software emulation of curses in System V Release 4.0, and more. It uses terminfo format, supports pads and color and multiple highlights and forms characters and function-key mapping, and has all the other SYSV-curses enhancements over BSD curses.

In mid-June 1995, the maintainer of 4.4BSD curses declared that he considered 4.4BSD curses obsolete, and is encouraging the keepers of Unix releases such as BSD/OS, freeBSD and netBSD to switch over to ncurses.

The ncurses code was developed under GNU/Linux. It has been in use for some time with OpenBSD as the system curses library, and on FreeBSD and NetBSD as an external package. It should port easily to any ANSI/POSIX-conforming UNIX. It has even been ported to OS/2 Warp!

The distribution includes the library and support utilities, including a terminfo compiler tic(1), a decompiler infocmp(1), clear(1), tput(1), tset(1), and a termcap conversion tool captoinfo(1). Full manual pages are provided for the library and tools.

The ncurses distribution is available via anonymous FTP at the GNU distribution site [1]ftp://ftp.gnu.org/gnu/ncurses/ .

It is also available at [2]ftp://invisible-island.net/ncurses/ .

Release Notes

This release is designed to be upward compatible from ncurses 5.0 and 5.3; very few applications will require recompilation, depending on the platform. These are the highlights from the change-log since ncurses 5.3 release.

Interface changes:

* add the remaining functions for X/Open curses wide-character support.

+ pecho_wchar()

+ slk_wset()

These are only available if the library is configured using the --enable-widec option.

* write getyx() and related 2-return macros in terms of getcury(), getcurx(), etc.

* simplify ifdef for bool declaration in curses.h

* modify ifdef's in curses.h that disabled use of __attribute__()

for g++, since recent versions implement the cases which ncurses uses.

* add key_defined() function, to tell which keycode a string is bound to.

New features and improvements:

* library

+ implement logic in lib_mouse.c to handle position reports which are generated when XFree86 xterm is initialized with

private modes 1002 or 1003. These are returned to the application as the REPORT_MOUSE_POSITION mask, which was not implemented.

+ modify soft-key initialization to use A_REVERSE if A_STANDOUT would not be shown when colors are used, i.e., if ncv#1 is set in the terminfo as is done in "screen".

+ add configure option for FreeBSD sysmouse, --with-sysmouse, and implement library support for that.

* programs:

+ tack:

o allow it to run from fallback terminfo data.

o reset colors after each color test, correct a place

where exit_standout_mode was used instead of exit_attribute_mode.

o improve bce test by making it set colors other than black on white.

+ tic:

o handle a case where an entry has no description, and capabilities begin on the same line as the entry name.

o allow a terminfo entry with a leading 2-character name.

o improved warnings when suppressing items to fit in termcap's 1023-byte limit.

o add check for multiple "tc=" clauses in a termcap.

o correct logic for resolving "use=" clauses allow infocmp and tic to show cancelled capabilities.

o check for incomplete line-drawing character mapping.

o check for missing/empty/illegal terminfo name.

+ tput:

o modify tput to use the same parameter analysis as tparm() does, to provide for user-defined strings, e.g.,

for xterm title, a corresponding capability might be title=\E]2;%p1%s^G,

+ tset:

o use the system's default values for CKILL and other default control characters.

o correct interchanged defaults for kill and interrupt characters, which caused it to report unnecessarily.

Major bug fixes:

* prevent recursion in wgetch() via wgetnstr() if the connection cannot be switched between cooked/raw modes because it is not a TTY.

* correct a case in _nc_remove_string(), used by define_key(), to avoid infinite loop if the given string happens to be a substring of other strings which are assigned to keys.

* modify logic of acsc to use the original character if no mapping is defined, rather than a null.

* several improvements for handling multi-cell display of wide characters.

+ modify setcchar() to allow converting control characters to complex characters.

+ correct handling of multibyte characters in waddch_literal() which force wrapping because they are started too late on the line.

+ modify setcchar() to allow for wchar_t input strings that have more than one spacing character.

* other fixes for wide-character support:

+ rewrote lib_acs.c to allow PutAttrChar() to decide how to render alternate-characters, i.e., to work with Linux console

and UTF-8 locale.

+ implement a workaround so that line-drawing works with screen's crippled UTF-8 support (tested with 3.9.13). This

only works with the wide-character support (--enable-widec); the normal library will simply suppress line-drawing when

running in a UTF-8 locale in screen.

+ corrections to lib_get_wstr.c:

o null-terminate buffer passed to setcchar(), which occasionally failed.

o map special characters such as erase- and kill-characters into key-codes so those will work as expected even if they are not mentioned in the terminfo.

* modify setupterm() to check if the terminfo and terminal-modes have already been read. This ensures that it does not reinvoke def_prog_mode() when an application calls more than one function, such as tgetent() and initscr().

* fix form_driver() cases for REQ_CLR_EOF, REQ_CLR_EOL, REQ_DEL_CHAR, REQ_DEL_PREV and REQ_NEW_LINE, which did not ensure the cursor was at the editing position before making modifications.

* correct keybound(), which reported definitions in the wrong table, i.e., the list of definitions which are disabled by keyok().

* fixes related to safe_sprintf.c:

+ correct an allocation size in safe_sprintf.c for the "*" format code.

+ correct safe_sprintf.c to not return a null pointer if the format happens to be an empty string.

+ make return value from _nc_printf_string() consistent. Before, depending on whether --enable-safe-sprintf was used,

it might not be cached for reallocating.

* other low-level improvements to the optimization code include:

+ if the output is a socket or other non-tty device, use 1 millisecond for the cost in mvcur; previously it was 9

milliseconds because the baudrate was not known.

+ modify lib_getch.c to avoid recursion via wgetnstr() when the input is not a tty and consequently mode-changes do not work.

+ fix several places in tack/pad.c which tested and used the parameter- and parameterless strings inconsistently.

+ change several tputs() calls in scrolling code to use putp(), to enable padding which may be needed for some terminals.

+ improve mvcur() by checking if it is safe to move when video attributes are set (msgr), and if not, reset/restore

attributes within that function rather than doing it separately in the GoTo() function in tty_update.c.

+ use tputs() rather than putp() in a few cases in tty_update.c since the corresponding delays are proportional to the number of lines affected: repeat_char, clr_eos, change_scroll_region.

* correct line/column reference in adjust_window(), needed to make special windows such as curscr track properly when resizing.

* fix a potential recursion between napms() and _nc_timed_wait()

* rewrote lib_insch.c, combining it with lib_insstr.c so both handle tab and other control characters consistently.

* do not try to open gpm mouse driver if standard output is not a tty; the gpm library does not make this check.

Portability:

* configure script:

+ new options:

--with-abi-version option.

addresses platforms where packagers have diverged from ncurses ABI numbering.

--with-manpage-format=catonly

addresses behavior of BSDI, allow install of man+cat files on NetBSD, whose behavior has

diverged by requiring both to be present.

--with-manpage-aliases

extends "--with-manpage-aliases" to provide the option of generating ".so" files rather than

symbolic links for manpage aliases.

--with-rel-version

workaround to force libtool on Darwin generate the "same" library names as with the "--with-shared"

option. The Darwin ld program does not work well with a zero as the minor-version value.

--with-trace

simplifies defining TRACE to incorporate trace() in libraries.

+ fixes/improvements for cross-compiling:

o allow BUILD_CC and related configure script variables to be overridden from the environment.

o use AC_CHECK_TOOL to get proper values for AR and LD for cross compiling.

o use $cross_compiling variable in configure script rather than comparing $host_alias and $target alias, since

"host" is traditionally misused in autoconf to refer to the target platform.

o modify run_tic.in to avoid using wrong shared libraries when cross-compiling.

+ fixes for Mac OS X:

o fix a redefinition of $RANLIB in the configure script when libtool is used.

o modify MKlib_gen.sh to avoid passing "#" tokens through the C preprocessor. This works around Mac OS X's

preprocessor, which insists on adding a blank on each side of the token.

+ workarounds for broken tools:

o add configure check for wchar_t and wint_t types, rather than rely on preprocessor definitions. Also work around

for gcc fixinclude bug which creates a shadow copy of curses.h if it sees these symbols apparently typedef'd.

o modify configure script to omit -Winline for gcc 3.3, since that feature is broken.

o several script fixes to work around the ironically named POSIXLY_CORRECT feature of GNU sed 4.0.

o modify configure script to avoid using "head -1", which does not work if POSIXLY_CORRECT (sic) is set.

o update configure script to reflect fix for AC_PROG_GCC_TRADITIONAL, which is broken in autoconf

2.5x for Mac OS X 10.2.3.

o repair check for missing C++ compiler, which is broken in autoconf 2.5x by hardcoding it to g++.

+ corrected ifdef's relating to configure check for wchar_t, etc.

+ remove configure script check to allow -Wconversion for older versions of gcc

+ modify configure script to accommodate libtool 1.5, as well as add an parameter to the "--with-libtool" option which can specify the pathname of libtool.

+ change several sed scripts to avoid using "\+" since it is not a BRE (basic regular expression). One instance caused

terminfo.5 to be misformatted on FreeBSD.

+ use '%' as sed substitute delimiter in run_tic script to avoid problems with pathname delimiters such as ':' and '@'.

+ add -D_XOPEN_SOURCE=500 if needed when configuring with "--enable-widec", to get mbstate_t declaration on HPUX 11.11.

* library:

+ adjust include-options in CF_ETIP_DEFINES to avoid missing ncurses_dll.h, fixing special definitions that may be needed for etip.h.

+ modify CF_LIB_SUFFIX for Itanium releases of HP-UX, which use a ".so" suffix.

+ improve ifdef's to guard against redefinition of wchar_t and wint_t in curses.h.

+ remove an #undef for KEY_EVENT from curses.tail used in the experimental NCURSES_WGETCH_EVENTS feature. The #undef confuses Debian dpkg's build script.

Features of Ncurses

The ncurses package is fully compatible with SVr4 (System V Release 4) curses:

* All 257 of the SVr4 calls have been implemented (and are documented).

* Full support for SVr4 curses features including keyboard mapping, color, forms-drawing with ACS characters, and automatic recognition of keypad and function keys.

* An emulation of the SVr4 panels library, supporting a stack of windows with backing store, is included.

* An emulation of the SVr4 menus library, supporting a uniform but flexible interface for menu programming, is included.

* An emulation of the SVr4 form library, supporting data collection through on-screen forms, is included.

* Binary terminfo entries generated by the ncurses tic(1) implementation are bit-for-bit-compatible with the entry format

SVr4 curses uses.

* The utilities have options to allow you to filter terminfo entries for use with less capable curses/terminfo versions such as the HP/UX and AIX ports.

The ncurses package also has many useful extensions over SVr4:

* The API is 8-bit clean and base-level conformant with the X/OPEN curses specification, XSI curses (that is, it implements all BASE level features, but not all EXTENDED features). Most

EXTENDED-level features not directly concerned with wide-character support are implemented, including many function calls not supported under SVr4 curses (but portability of all calls is documented so you can use the SVr4 subset only).

* Unlike SVr3 curses, ncurses can write to the rightmost-bottommost corner of the screen if your terminal has an insert-character capability.

* Ada95 and C++ bindings.

* Support for mouse event reporting with X Window xterm and OS/2 console windows.

* Extended mouse support via Alessandro Rubini's gpm package.

* The function wresize() allows you to resize windows, preserving their data.

* The function use_default_colors() allows you to use the terminal's default colors for the default color pair, achieving the effect of transparent colors.

* The functions keyok() and define_key() allow you to better control the use of function keys, e.g., disabling the ncurses KEY_MOUSE, or by defining more than one control sequence to map to a given

key code.

* Support for 16-color terminals, such as aixterm and XFree86 xterm.

* Better cursor-movement optimization. The package now features a cursor-local-movement computation more efficient than either BSD's or System V's.

* Super hardware scrolling support. The screen-update code incorporates a novel, simple, and cheap algorithm that enables it to make optimal use of hardware scrolling, line-insertion, and

line-deletion for screen-line movements. This algorithm is more powerful than the 4.4BSD curses quickch() routine.

* Real support for terminals with the magic-cookie glitch. The screen-update code will refrain from drawing a highlight if the magic- cookie unattributed spaces required just before the

beginning and after the end would step on a non-space character. It will automatically shift highlight boundaries when doing so would make it possible to draw the highlight without changing the visual appearance of the screen.

* It is possible to generate the library with a list of pre-loaded fallback entries linked to it so that it can serve those terminal types even when no terminfo tree or termcap file is accessible (this may be useful for support of screen-oriented programs that must run in single-user mode).

* The tic(1)/captoinfo utility provided with ncurses has the ability to translate many termcaps from the XENIX, IBM and AT&T extension sets.

* A BSD-like tset(1) utility is provided.

* The ncurses library and utilities will automatically read terminfo entries from $HOME/.terminfo if it exists, and compile to that directory if it exists and the user has no write access to the

system directory. This feature makes it easier for users to have personal terminfo entries without giving up access to the system terminfo directory.

* You may specify a path of directories to search for compiled descriptions with the environment variable TERMINFO_DIRS (this generalizes the feature provided by TERMINFO under stock System V.)

* In terminfo source files, use capabilities may refer not just to other entries in the same source file (as in System V) but also to compiled entries in either the system terminfo directory or the user's $HOME/.terminfo directory.

* A script (capconvert) is provided to help BSD users transition from termcap to terminfo. It gathers the information in a TERMCAP environment variable and/or a ~/.termcap local entries file and converts it to an equivalent local terminfo tree under

$HOME/.terminfo.

* Automatic fallback to the /etc/termcap file can be compiled in when it is not possible to build a terminfo tree. This feature is neither fast nor cheap, you don't want to use it unless you have to, but it's there.

* The table-of-entries utility toe makes it easy for users to see exactly what terminal types are available on the system.

* The library meets the XSI requirement that every macro entry point have a corresponding function which may be linked (and will be prototype-checked) if the macro definition is disabled with #undef.

* An HTML "Introduction to Programming with NCURSES" document provides a narrative introduction to the curses programming interface.

State of the Package

Numerous bugs present in earlier versions have been fixed; the library is far more reliable than it used to be. Bounds checking in many `dangerous' entry points has been improved. The code is now type-safe according to gcc -Wall. The library has been checked for malloc leaks and arena corruption by the Purify memory-allocation tester.

The ncurses code has been tested with a wide variety of applications

including (versions starting with those noted):

cdk

Curses Development Kit

[3]http://invisible-island.net/cdk/

[4]http://www.vexus.ca/products/CDK/

ded

directory-editor

[5]http://invisible-island.net/ded/

dialog

the underlying application used in Slackware's setup, and the basis for similar applications on GNU/Linux.

[6]http://invisible-island.net/dialog/

lynx

the character-screen WWW browser

[7]http://lynx.isc.org/release/

Midnight Commander file manager

[8]http://www.ibiblio.org/mc/

mutt

mail utility

[9]http://www.mutt.org/

ncftp

file-transfer utility

[10]http://www.ncftp.com/

nvi

New vi versions 1.50 are able to use ncurses versions 1.9.7 and later.

[11]http://www.bostic.com/vi/

pinfo

Lynx-like info browser.

[12]http://dione.ids.pl/~pborys/software/pinfo/

tin

newsreader, supporting color, MIME [13]http://www.tin.org/

vh-1.6

Volks-Hypertext browser for the Jargon File

as well as some that use ncurses for the terminfo support alone:

minicom

terminal emulator

[15]http://www.netsonic.fi/~walker/minicom.html

vile

vi-like-emacs

[16]http://invisible-island.net/vile/

The ncurses distribution includes a selection of test programs

(including a few games).

Who's Who and What's What

Zeyd Ben-Halim started it from a previous package pcurses, written by Pavel Curtis. Eric S. Raymond continued development. Jürgen Pfeifer wrote most of the form and menu libraries. Ongoing work is being done by [17]Thomas Dickey. Thomas Dickey acts as the maintainer for the Free Software Foundation, which holds the copyright on ncurses.

Contact the current maintainers at [18]bug-ncurses@gnu.org.

To join the ncurses mailing list, please write email to bug-ncurses-request@gnu.org containing the line:

subscribe <name>@<host.domain>

This list is open to anyone interested in helping with the development and testing of this package.

Beta versions of ncurses and patches to the current release are made available at [19]ftp://invisible-island.net/ncurses/ .

Future Plans

* Extended-level XPG4 conformance, with internationalization support.

* Ports to more systems, including DOS and Windows.

We need people to help with these projects. If you are interested in working on them, please join the ncurses list.

Other Related Resources

The distribution provides a newer version of the terminfo-format terminal description file maintained by [20]Eric Raymond . Unlike the older version, the termcap and terminfo data are provided in the same file.

You can find lots of information on terminal-related topics not covered in the terminfo file at [21]Richard Shuford's archive .

References

1. ftp://ftp.gnu.org/gnu/ncurses/

2. ftp://invisible-island.net/ncurses/

3. http://invisible-island.net/cdk/

4. http://www.vexus.ca/products/CDK/

5. http://invisible-island.net/ded/

6. http://invisible-island.net/dialog/

7. http://lynx.isc.org/release/

8. http://www.ibiblio.org/mc/

9. http://www.mutt.org/

10. http://www.ncftp.com/

11. http://www.bostic.com/vi/

12. http://dione.ids.pl/~pborys/software/pinfo/

13. http://www.tin.org/

14. http://www.debian.org/Packages/unstable/text/vh.html

15. http://www.netsonic.fi/~walker/minicom.html

16. http://invisible-island.net/vile/

17. mailto:dickey@invisible-island.net

18. mailto:bug-ncurses@gnu.org

19. ftp://invisible-island.net/ncurses/

20. http://www.catb.org/~esr/terminfo/

21. http://www.cs.utk.edu/~shuford/terminal_index.html