pkg_select is a tool to navigate pkgsrc

- pkg_select
  - is coded in C
  - follows /usr/share/misc/style specifications
  - has no other packages dependencies
  - is linked only to curses(3) and libc

/usr/pkg/bin/pkg_select:
  -lcurses.6 => /usr/lib/libcurses.so.6
  -lc.12 => /usr/lib/libc.so.12
What is pkg_select?

- frontend to the pkgsrc packaging system.
- simple curse interface
- fetch pkgsrc if not present in the system
  - from a local directory
  - from CVS
  - from ftp
- upgrade pkgsrc from CVS
- browse pkgsrc hierarchy
  - from a local pkgsrc
  - from a distant ftp pkgsrc repository
  - from a local binary repository
- browse pkg_dbdir and show installed packages
- gather various informations about packages
- build, install, upgrade a source package
- install a binary package
At startup, if it has been called without any special flags, pkg_select will check if a pkgsrc hierarchy is available at:

- `${PKGSRCDIR}` or
- a path specified by `-b`, or
- `/usr/pkgsrc` by default.
No pkgsrc present

pkgsrc was not found in /tmp/inexistant, download it? [Y/n]
Fetching

If not present, it will ask you to download pkgsrc (pkgsrc.c)
- from a ftp mirror list
- from a CVS mirror list
- from a local copy by specifying a directory
"Classic" local pkgsrc mode

- reads the directory containing pkgsrc using scandir(3) (parsedirs.c)
- reads every category Makefile and prints them associated with the COMMENT variable (makefile.c)

From the browsing interface the following actions may be done:

- Update pkgsrc by CVS (pkgsrc.c)
- Find one or more packages (parsedirs.c) and browse the result just as if you were browsing pkgsrc

continues next slide...
<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>archivers</td>
<td>Archivers</td>
</tr>
<tr>
<td>audio</td>
<td>Audio tools</td>
</tr>
<tr>
<td>benchmarks</td>
<td>Benchmarking tools</td>
</tr>
<tr>
<td>biology</td>
<td>Software for the biological sciences</td>
</tr>
<tr>
<td>cad</td>
<td>CAD tools</td>
</tr>
<tr>
<td>chat</td>
<td>Communication programs</td>
</tr>
<tr>
<td>comms</td>
<td>Communication utilities</td>
</tr>
<tr>
<td>converters</td>
<td>Document format and character code converters</td>
</tr>
<tr>
<td>crosspkgtool</td>
<td>Tools for cross-building pkgsrc</td>
</tr>
<tr>
<td>databases</td>
<td>Databases</td>
</tr>
<tr>
<td>devel</td>
<td>Development utilities</td>
</tr>
<tr>
<td>editors</td>
<td>Editors</td>
</tr>
<tr>
<td>emulators</td>
<td>Emulators for other operating systems</td>
</tr>
<tr>
<td>finance</td>
<td>Monetary, financial and related applications</td>
</tr>
<tr>
<td>fonts</td>
<td>Fonts</td>
</tr>
<tr>
<td>games</td>
<td>Games</td>
</tr>
<tr>
<td>geography</td>
<td>Software for geographical-related uses</td>
</tr>
</tbody>
</table>

**Commands:**
- [ret] enter
- [<-] back
- [q] quit
- [/] search
- [n] next
- [o] other cmds
- [f] pkgfind
- [l] installed packages
"Classic" local pkgsrc mode

- Browse installed packages (via pkg_dbdir), same as above
- Tag / Untag one or more packages for mass-installation process (install_many.c)
- enter preferences menu (prefs.c)
## Installed Packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6tunnel</td>
<td>v4/v6 protocol translation</td>
</tr>
<tr>
<td>GConf2</td>
<td>Configuration database system used by GNOME (v2)</td>
</tr>
<tr>
<td>ImageMagick</td>
<td>Package for display and interactive manipulation</td>
</tr>
<tr>
<td>MesaLib</td>
<td>Graphics library similar to SGI's OpenGL</td>
</tr>
<tr>
<td>ORBit</td>
<td>High-performance CORBA ORB with support for the C.</td>
</tr>
<tr>
<td>ORBit2</td>
<td>High-performance CORBA ORB with C language support</td>
</tr>
<tr>
<td>Xcomposite</td>
<td>Xcomposite extension (library)</td>
</tr>
<tr>
<td>Xfixes</td>
<td>X Fixes extension of X RandR</td>
</tr>
<tr>
<td>Xft2</td>
<td>Library for configuring and customizing font acc.</td>
</tr>
<tr>
<td>Xrandr</td>
<td>X RandR extension (Library)</td>
</tr>
<tr>
<td>Xrender</td>
<td>Client library for the X Rendering Extension prot.</td>
</tr>
<tr>
<td>a2ps</td>
<td>Formats an ascii file for printing on a postscript</td>
</tr>
<tr>
<td>aalib</td>
<td>ASCII Art library</td>
</tr>
<tr>
<td>aalib-x11</td>
<td>ASCII Art library with X11 support</td>
</tr>
<tr>
<td>amule</td>
<td>Another Emule P2P client</td>
</tr>
<tr>
<td>apcupsd</td>
<td>UPS power management for APCC Products</td>
</tr>
<tr>
<td>aspell</td>
<td>Spell checker with good multi-language support</td>
</tr>
<tr>
<td>asterisk</td>
<td>The Asterisk Software PBX</td>
</tr>
</tbody>
</table>

[ret] enter  [/] search  [n] next  [o] other cmds  
"Classic" local pkgsrc mode

When a DESCR file is present, pkg_select enters a package administration and information mode (*pkg_info.c*)
"Classic" local pkgsrc mode

Information is gathered from package **Makefile**

- Package description and file list are read from **DESCR** and **PLIST**.

- Package dependencies calculated via `make show-var VARNAME=DEPENDS`

- Enables browse dependencies just like in pkgsrc

- Build and install executes: `make install clean`

- Build and upgrade executes: `make update clean`

- Install binary package

- Deinstall a package
pkgsrc-over-ftp mode

When started with the -l flag, pkg_select enters the "live pkgsrc" mode (live.c).
Using pkg_select this way:
- you don't have to have a local pkgsrc hierarchy
- connects to a ftp pkgsrc mirror
- behaves just as it was browsing a local repository,
- uses slightly modified versions of pkg_install's ftpio.c functions (ftpio.c).
pkgsrc-over-ftp mode
pkgsrc-over-ftp mode

As reading remote Makefiles can take lot of time if you are browsing through a slow link

the -m flag will prevent pkg_select to read them.

In this mode:

only categories and package names are shown.

This can also be set by the live_ftp_read_makefiles configuration variable.
pkgsrc-over-ftp mode
As in classic operation mode, when a DESCR file is found, `pkg_select` shows the package administration / information screen.

In live-ftp mode, fewer informations are shown, and only binary installation can be done (i.e. no build).
local binary repository mode

A feature that could be used as a post-installation process so binary packages installation is possible without local pkgsrc hierarchy

Available at the pkg_select CVS repository:

:pserver:anoncvs@gcu.info:/cvs

If PACKAGES points to a directory containing binary packages:
- pkg_select will build a virtual pkgsrc hierarchy
- pkg_select creates package category directory, package directory, Makefile, DESCR and PLIST
- This virtual hierarchy is then used as a classic pkgsrc, with fewer informations and only binary installation.
local binary repository mode
Upcoming plans

- Known bugs
  - local cp is reported to fail sometimes plus print garbage
  - system binaries installation seems to be leaked
  - packages Makefile with lots of inclusions seem to lead to SIGSEGV
- install_many.c needs work, at last on presentation
- Give local cp availability to extract pkgsrc from compressed archive
- Add an interface to modify mk.conf, probably using packages options
- Rewrite configuration file screen / handling (ugly by now)
- pkg_select has been tested under NetBSD/x86, FreeBSD/x86 and DragonFly/x86, many more platforms must be tested.
- Hubert requested a more "friendly" menu-driven interface, from which you could go to the browsing screen, something ala pine may be nice.
Thanks to the GCU [http://gcu-squad.org](http://gcu-squad.org)

Thanks to the NetBSD project

Questions ?