ptx

The GNU permuted indexer
Edition 0.3, for ptx version 0.3
November 1993

by Francois Pinard

1 Introduction

This is the 0.3 beta release of `ptx`, the GNU version of a permuted index generator. This software has the main goal of providing a replacement for the traditional `ptx` as found on System V machines, able to handle small files quickly, while providing a platform for more development.

This version reimplements and extends traditional `ptx`. Among other things, it can produce a readable *KWIC* (keywords in their context) without the need of `nroff`, there is also an option to produce TEX compatible output. This version does not handle huge input files, that is, those files which do not fit in memory all at once.

Please note that an overall renaming of all options is foreseeable. In fact, GNU ptx specifications are not frozen yet.

2 How to use this program

This tool reads a text file and essentially produces a permuted index, with each keyword in its context. The calling sketch is one of:

```
ptx [option ...] [file ...]
```

or:

```
ptx -G [option ...] [input [output]]
```

The '`-G`' (or its equivalent: '`--traditional`') option disables all GNU extensions and revert to traditional mode, thus introducing some limitations, and changes several of the program's default option values. When '`-G`' is not specified, GNU extensions are always enabled. GNU extensions to `ptx` are documented wherever appropriate in this document. See See Chapter 3 [Compatibility], page 8 for an explicit list of them.

Individual options are explained later in this document.

When GNU extensions are enabled, there may be zero, one or several *file* after the options. If there is no *file*, the program reads the standard input. If there is one or several *file*, they give the name of input files which are all read in turn, as if all the input files were concatenated. However, there is a full contextual break between each file and, when automatic referencing is requested, file names and line numbers refer to individual text input files. In all cases, the program produces the permuted index onto the standard output.

When GNU extensions are *not* enabled, that is, when the program operates in traditional mode, there may be zero, one or two parameters besides the options. If there is no parameters, the program reads the standard input and produces the permuted index onto the standard output. If there is only one parameter, it names the text *input* to be read instead of the standard input. If two parameters are given, they give respectively the name of the *input* file to read and the name of the *output* file to produce. *Be very careful* to note that, in this case, the contents of file given by the second parameter is destroyed. This behaviour is dictated only by System V `ptx` compatibility, because GNU Standards discourage output parameters not introduced by an option.

Note that for *any* file named as the value of an option or as an input text file, a single dash `-` may be used, in which case standard input is assumed. However, it would not make sense to use this convention more than once per program invocation.

2.1 General options

`-C`
`--copyright`
: Prints a short note about the Copyright and copying conditions, then exit without further processing.

`-G`
`--traditional`
: As already explained, this option disables all GNU extensions to `ptx` and switch to traditional mode.

`--help` Prints a short help on standard output, then exit without further processing.

`--version`
: Prints the program verison on standard output, then exit without further processing.

2.2 Charset selection

As it is setup now, the program assumes that the input file is coded using 8-bit ISO 8859-1 code, also known as Latin-1 character set, *unless* if it is compiled for MS-DOS, in which case it uses the character set of the IBM-PC. (GNU `ptx` is not known to work on smaller MS-DOS machines anymore.) Compared to 7-bit ASCII, the set of characters which are letters is then different, this fact alters the behaviour of regular expression matching. Thus, the default regular expression for a keyword allows foreign or diacriticized letters. Keyword sorting, however, is still crude; it obeys the underlying character set ordering quite blindly.

`-f`
`--ignore-case`
: Fold lower case letters to upper case for sorting.

2.3 Word selection

`-b` *file*

`--break-file=`*file*
: This option is an alternative way to option `-W` for describing which characters make up words. This option introduces the name of a file which contains a list of characters which can*not* be part of one word, this file is called the *Break file*. Any character which is not part of the Break file is a word constituent. If both options `-b` and `-W` are specified, then `-W` has precedence and `-b` is ignored.

  When GNU extensions are enabled, the only way to avoid newline as a break character is to write all the break characters in the file with no newline at all, not even at the end of the file. When GNU extensions are disabled, spaces, tabs and newlines are always considered as break characters even if not included in the Break file.

`-i` *file*
`--ignore-file=`*file*
: The file associated with this option contains a list of words which will never be taken as keywords in concordance output. It is called the *Ignore file*. The file contains exactly one word in each line; the end of line separation of words is not subject to the value of the `-S` option.

  There is a default Ignore file used by `ptx` when this option is not specified, usually found in '`/usr/local/lib/eign`' if this has not been changed at installation time. If you want to deactivate the default Ignore file, specify `/dev/null` instead.

`-o` *file*
`--only-file=`*file*
: The file associated with this option contains a list of words which will be retained in concordance output, any word not mentioned in this file is ignored. The file

is called the *Only file*. The file contains exactly one word in each line; the end of line separation of words is not subject to the value of the `-S` option.

There is no default for the Only file. In the case there are both an Only file and an Ignore file, a word will be subject to be a keyword only if it is given in the Only file and not given in the Ignore file.

`-r`
`--references`

On each input line, the leading sequence of non white characters will be taken to be a reference that has the purpose of identifying this input line on the produced permuted index. See See Section 2.4 [Output formatting], page 5 for more information about reference production. Using this option change the default value for option `-S`.

Using this option, the program does not try very hard to remove references from contexts in output, but it succeeds in doing so *when* the context ends exactly at the newline. If option `-r` is used with `-S` default value, or when GNU extensions are disabled, this condition is always met and references are completely excluded from the output contexts.

`-S` *regexp*
`--sentence-regexp=`*regexp*

This option selects which regular expression will describe the end of a line or the end of a sentence. In fact, there is other distinction between end of lines or end of sentences than the effect of this regular expression, and input line boundaries have no special significance outside this option. By default, when GNU extensions are enabled and if `-r` option is not used, end of sentences are used. In this case, the precise *regex* is imported from GNU emacs:

```
[.?!][]\"')}]*\\($\\|\t\\|  \\)[ \t\n]*
```

Whenever GNU extensions are disabled or if `-r` option is used, end of lines are used; in this case, the default *regexp* is just:

```
\n
```

Using an empty REGEXP is equivalent to completely disabling end of line or end of sentence recognition. In this case, the whole file is considered to be a single big line or sentence. The user might want to disallow all truncation flag generation as well, through option `-F ""`. See section "Syntax of Regular Expressions" in *The GNU Emacs Manual*.

When the keywords happen to be near the beginning of the input line or sentence, this often creates an unused area at the beginning of the output context line; when the keywords happen to be near the end of the input line or sentence, this often creates an unused area at the end of the output context line. The program tries to fill those unused areas by wrapping around context in them; the tail of the input line or sentence is used to fill the unused area on the left of the output line; the head of the input line or sentence is used to fill the unused area on the right of the output line.

As a matter of convenience to the user, many usual backslashed escape sequences, as found in the C language, are recognized and converted to the corresponding characters by `ptx` itself.

`-W` *regexp*
`--word-regexp=`*regexp*

This option selects which regular expression will describe each keyword. By default, if GNU extensions are enabled, a word is a sequence of letters; the *regexp* used is `\w+`. When GNU extensions are disabled, a word is by default anything which ends with a space, a tab or a newline; the *regexp* used is `[^ \t\n]+`.

An empty REGEXP is equivalent to not using this option, letting the default dive in. See section "Syntax of Regular Expressions" in *The GNU Emacs Manual*.

As a matter of convenience to the user, many usual backslashed escape sequences, as found in the C language, are recognized and converted to the corresponding characters by `ptx` itself.

2.4 Output formatting

Output format is mainly controlled by `-O` and `-T` options, described in the table below. When neither `-O` nor `-T` is selected, and if GNU extensions are enabled, the program choose an output format suited for a dumb terminal. Each keyword occurrence is output to the center of one line, surrounded by its left and right contexts. Each field is properly justified, so the concordance output could readily be observed. As a special feature, if automatic references are selected by option `-A` and are output before the left context, that is, if option `-R` is *not* selected, then a colon is added after the reference; this nicely interfaces with GNU Emacs `next-error` processing. In this default output format, each white space character, like newline and tab, is merely changed to exactly one space, with no special attempt to compress consecutive spaces. This might change in the future. Except for those white space characters, every other character of the underlying set of 256 characters is transmitted verbatim.

Output format is further controlled by the following options.

`-g` *number*
`--gap-size=`*number*

Select the size of the minimum white gap between the fields on the output line.

`-w` *number*
`--width=`*number*

Select the output maximum width of each final line. If references are used, they are included or excluded from the output maximum width depending on the value of option `-R`. If this option is not selected, that is, when references are output before the left context, the output maximum width takes into account the maximum length of all references. If this options is selected, that is, when references are output after the right context, the output maximum width does not take into account the space taken by references, nor the gap that precedes them.

`-A`
`--auto-reference`

Select automatic references. Each input line will have an automatic reference made up of the file name and the line ordinal, with a single colon between them. However, the file name will be empty when standard input is being read. If both `-A` and `-r` are selected, then the input reference is still read and skipped, but the automatic reference is used at output time, overriding the input reference.

`-R`
`--right-side-refs`

In default output format, when option `-R` is not used, any reference produced by the effect of options `-r` or `-A` are given to the far right of output lines, after the right context. In default output format, when option `-R` is specified, references are rather given to the beginning of each output line, before the left context. For any other output format, option `-R` is almost ignored, except for the fact that the width of references is *not* taken into account in total output width given by `-w` whenever `-R` is selected.

This option is automatically selected whenever GNU extensions are disabled.

`-F` *string*
`--flac-truncation=`*string*

This option will request that any truncation in the output be reported using the string *string*. Most output fields theoretically extend towards the beginning or the end of the current line, or current sentence, as selected with option `-S`. But there is a maximum allowed output line width, changeable through option `-w`, which is further divided into space for various output fields. When a field has to be truncated because cannot extend until the beginning or the end of the current line to fit in the, then a truncation occurs. By default, the string used is a single slash, as in `-F /`.

string may have more than one character, as in `-F ...`. Also, in the particular case *string* is empty (`-F ""`), truncation flagging is disabled, and no truncation marks are appended in this case.

As a matter of convenience to the user, many usual backslashed escape sequences, as found in the C language, are recognized and converted to the corresponding characters by `ptx` itself.

`-M` *string*
`--macro-name=`*string*

Select another *string* to be used instead of '`xx`', while generating output suitable for `nroff`, `troff` or TEX.

`-O`
`--format=roff`

Choose an output format suitable for `nroff` or `troff` processing. Each output line will look like:

`.xx "`*tail*`" "`*before*`" "`*keyword_and_after*`" "`*head*`" "`*ref*`"`

so it will be possible to write an '`.xx`' roff macro to take care of the output typesetting. This is the default output format when GNU extensions are disabled. Option '`-M`' might be used to change '`xx`' to another macro name.

In this output format, each non-graphical character, like newline and tab, is merely changed to exactly one space, with no special attempt to compress consecutive spaces. Each quote character: `"` is doubled so it will be correctly processed by `nroff` or `troff`.

`-T`
`--format=tex`

Choose an output format suitable for TEX processing. Each output line will look like:

`\xx {`*tail*`}{`*before*`}{`*keyword*`}{`*after*`}{`*head*`}{`*ref*`}`

so it will be possible to write write a `\xx` definition to take care of the output typesetting. Note that when references are not being produced, that is, neither option `-A` nor option `-r` is selected, the last parameter of each `\xx` call is inhibited. Option '`-M`' might be used to change '`xx`' to another macro name.

In this output format, some special characters, like `$`, `%`, `&`, `#` and `_` are automatically protected with a backslash. Curly brackets `{`, `}` are also protected with a backslash, but also enclosed in a pair of dollar signs to force mathematical mode. The backslash itself produces the sequence `\backslash{}`. Circumflex and tilde diacritics produce the sequence `^\{ }` and `~\{ }` respectively. Other diacriticized characters of the underlying character set produce an appropriate TEX sequence as far as possible. The other non-graphical characters, like newline and tab, and all others characters which are not part of ASCII, are merely changed to exactly one space, with no special attempt to compress consecutive spaces. Let me know how to improve this special character processing for TEX.

3 The GNU extensions to ptx

This version of ptx contains a few features which do not exist in System V ptx. These extra features are suppressed by using the '-G' command line option, unless overridden by other command line options. Some GNU extensions cannot be recovered by overriding, so the simple rule is to avoid '-G' if you care about GNU extensions. Here are the differences between this program and System V ptx.

- This program can read many input files at once, it always writes the resulting concordance on standard output. On the other end, System V ptx reads only one file and produce the result on standard output or, if a second *file* parameter is given on the command, to that *file*.

  Having output parameters not introduced by options is a quite dangerous practice which GNU avoids as far as possible. So, for using ptx portably between GNU and System V, you should pay attention to always use it with a single input file, and always expect the result on standard output. You might also want to automatically configure in a '-G' option to ptx calls in products using ptx, if the configurator finds that the installed ptx accepts '-G'.
- The only options available in System V ptx are options '-b', '-f', '-g', '-i', '-o', '-r', '-t' and '-w'. All other options are GNU extensions and are not repeated in this enumeration. Moreover, some options have a slightly different meaning when GNU extensions are enabled, as explained below.
- By default, concordance output is not formatted for troff or nroff. It is rather formatted for a dumb terminal. troff or nroff output may still be selected through option -O.
- Unless -R option is used, the maximum reference width is subtracted from the total output line width. With GNU extensions disabled, width of references is not taken into account in the output line width computations.
- All 256 characters, even *NUL*s, are always read and processed from input file with no adverse effect, even if GNU extensions are disabled. However, System V ptx does not accept 8-bit characters, a few control characters are rejected, and the tilda ~ is condemned.
- Input line length is only limited by available memory, even if GNU extensions are disabled. However, System V ptx processes only the first 200 characters in each line.
- The break (non-word) characters default to be every character except all letters of the underlying character set, diacriticized or not. When GNU extensions are disabled, the break characters default to space, tab and newline only.
- The program makes better use of output line width. If GNU extensions are disabled, the program rather tries to imitate System V ptx, but still, there are some slight disposition glitches this program does not completely reproduce.
- The user can specify both an Ignore file and an Only file. This is not allowed with System V ptx.