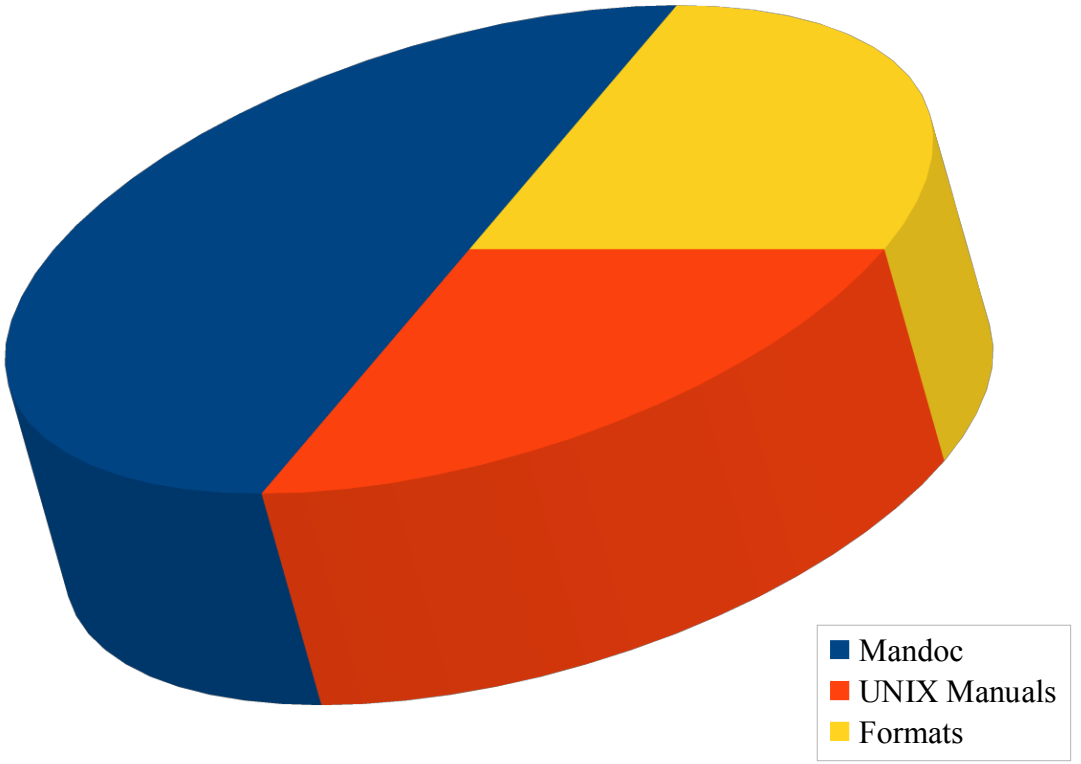
Ref: www.flickr.com/photos/jrubinic (Flickr Creative Commons Archive)

structure

BSDCan-2011

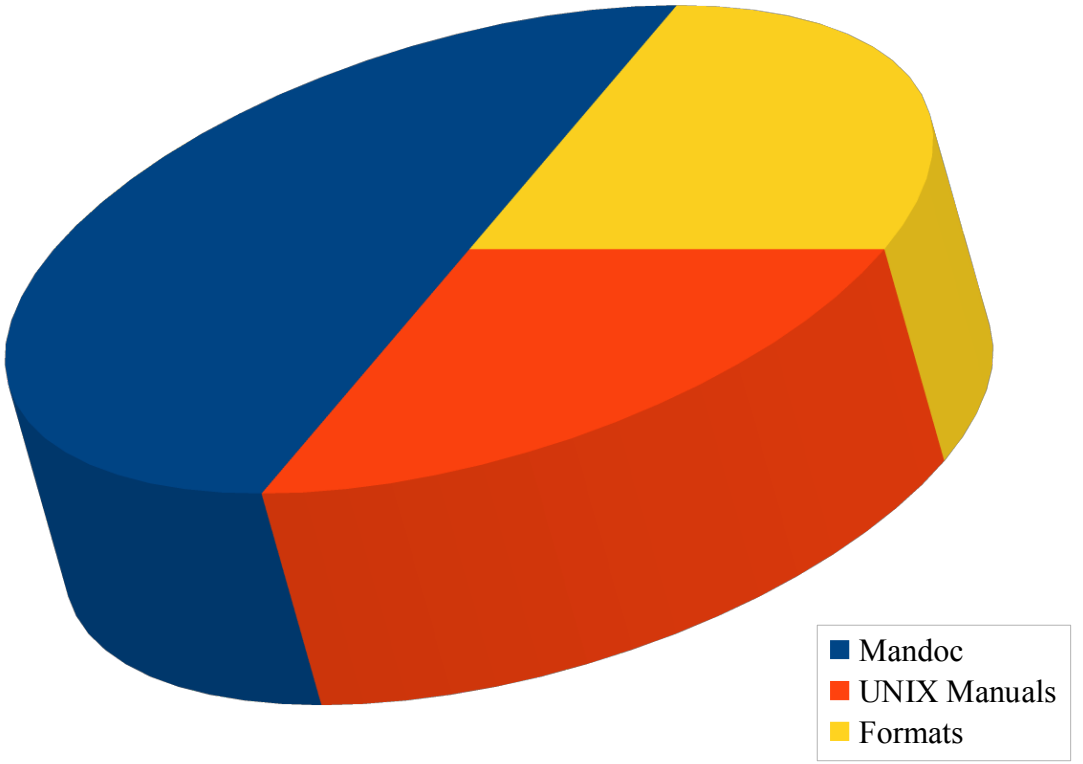
- UNIX Manuals
- Formats
- Mandoc
  http://mdocml.bsd.lv
  - mandoc.cgi
    http://mdocml.bsd.lv/mandoc-cgi



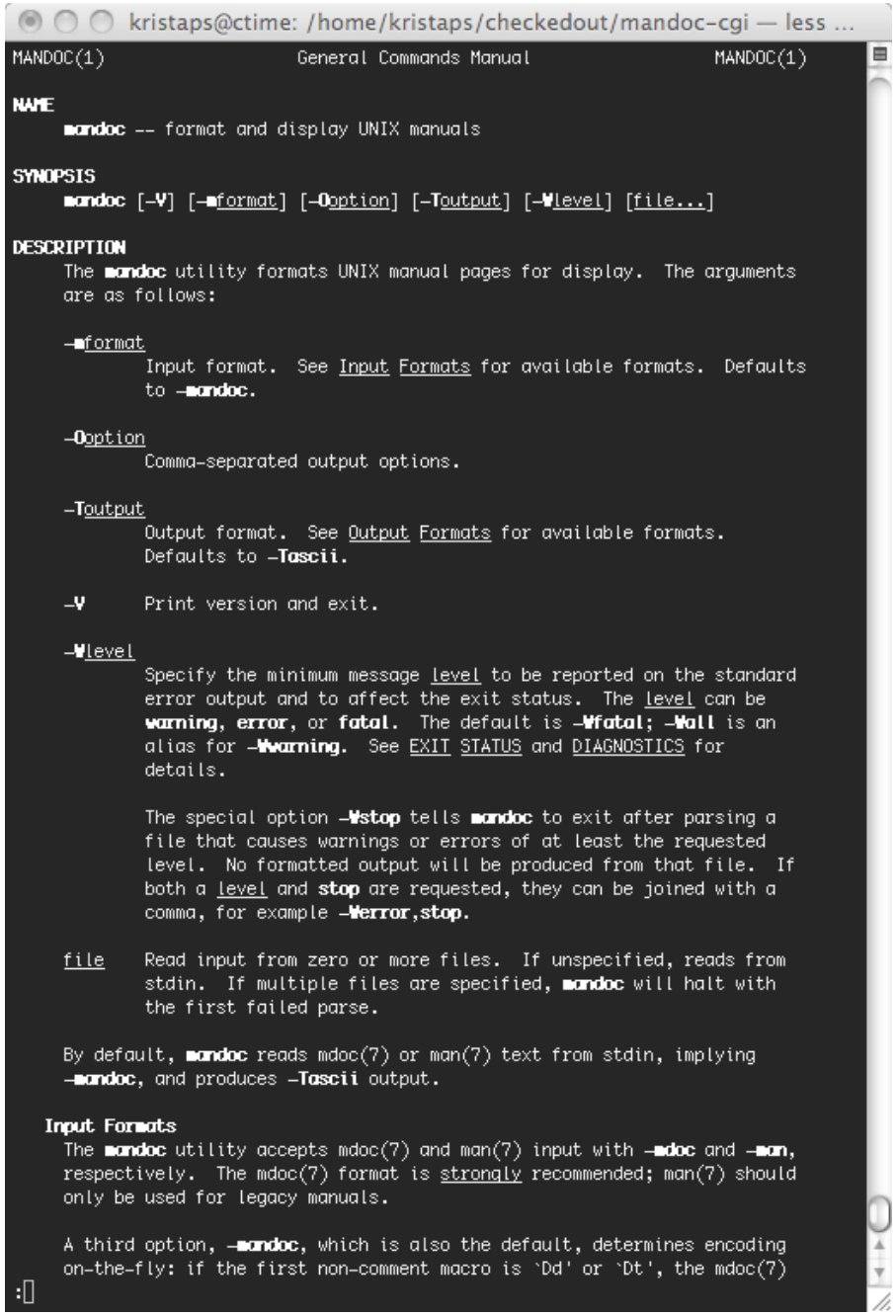
structure

BSDCan-2011

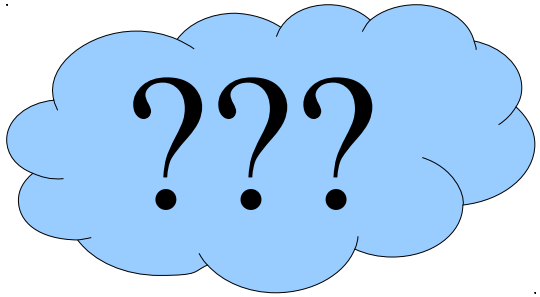
- UNIX Manuals
- Formats
- Mandoc
  http://mdocml.bsd.lv
  - mandoc.cgi
    http://mdocml.bsd.lv/mandoc-cgi



unix manuals – whatis



unix manuals – rcslog



unix manuals – rcslog RUNOFF

1960

1964: RUNOFF (CTSS)

1970

SUBJECT: TYPSET and RUNOFF, Memorandum editor and type-out commands.

DATE: November 6, 1964

[...]

“Control Words:

“Input generally consists of English text, 360 or fewer characters to a line. Control words must begin a new line, and begin with a period so that they may be distinguished from other text. RUNOFF does not print the control words.”

[...]

.space n

Insert "n" vertical spaces (carriage returns) in the copy. If "n" carries spaclng to the bottom of a page, spacing is stopped. If "n" is absent or 0, one space is inserted.

1980

1990

2000

2010

2011: BSDCan

Ref: http://mit.edu/Saltzer/www/publications/CC-244.html (TYPESET and RUNOFF)

unix manuals – rcslog runoff

1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)

1980

1990

2000

2010

2011: BSDCan

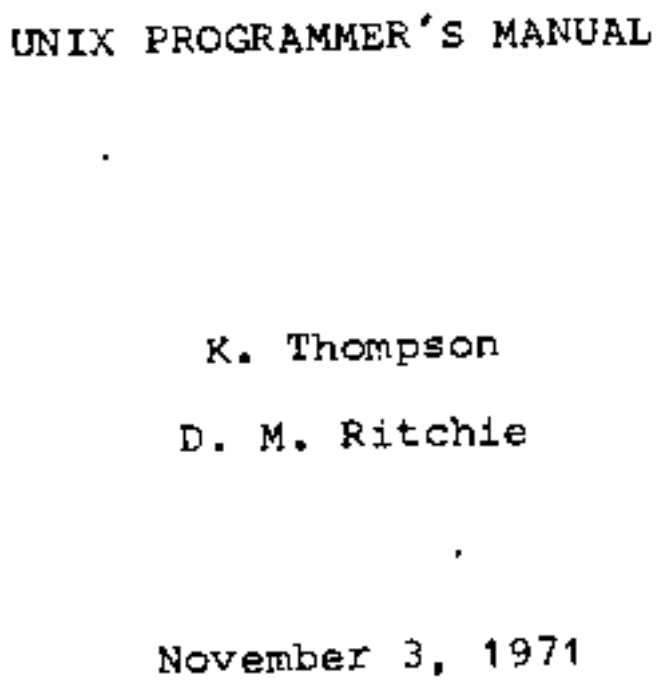
Ref: D. M. Ritchie, The Evolution of the UNIX Time-sharing System

1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)
1971: roff (UNIX, PDP-11)


UNIX PROGRAMMER'S MANUAL

K. Thompson

D. M. Ritchie

November 3, 1971

1980

1990

2000

2010

2011: BSDCan

Ref: http://cm.bell-labs.com/cm/cs/who/dmr/1stEdman.html (UNIX Programmer's Manuals, First Ed.)

This manual is divided into seven sections:

I. Commands
II. System calls
III. Subroutines
IV. Special files
V. File formats
VI. User-maintained programs
VII. Miscellaneous

Ref: http://www.openbsd.org (man.7, OpenBSD-current, 9-5-2011)
Ref: http://cm.bell-labs.com/cm/cs/who/dmr/1stEdman.html (UNIX Programmer's Manuals, First Ed.)

The name section repeats the entry name and gives a very short description of its purpose.

The synopsis summarizes the use of the program being described. A few conventions are used, particularly in the Commands section:

> Underlined words are considered literals, and are typed just as they appear.
>
> Square brackets ([]) around an argument indicate that the argument is optional. When an argument is given as "name", it always refers to a file name.
>
> Ellipses "..." are used to show that the previous argument-prototype may be repeated.
>
> A final convertion is used by the commands themselves. An argument beginning with a minus sign "-" is often taken to mean some sort of flag argument even if it appears in a position where a file name could appear. Therefore, it is unwise to have files whose names begin with "-".

The description section discusses in detail the subject at hand.

The files section gives the names of files which are built into the program.

A see also section gives pointers to related information.

A diagnostics section discusses the diagnostics that may be produced. This section tends to be as terse as the diagnostics themselves.

The bugs section gives known bugs and sometimes deficiencies. Occasionally also the suggested fix is described.

The owner section gives the name of the person or persons to be consulted in case of difficulty. The rule has been that the last one to modify something owns it, so the owner is not necessarily the author. The owner's initials stand for:

Ref: http://www.openbsd.org (man.7, OpenBSD-current, 9-5-2011)
Ref: http://cm.bell-labs.com/cm/cs/who/dmr/1stEdman.html (UNIX Programmer's Manuals, First Ed.)

unix manuals – rcslog nroff

1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)
1971: roff (UNIX, PDP-11)
1973: nroff, troff (UNIX, PDP-11)

1980

1990

2000

2010

2011: BSDCan

Ref: http://heirloom.sourceforge.net/doctools.html (Heirloom Documentation Tools)

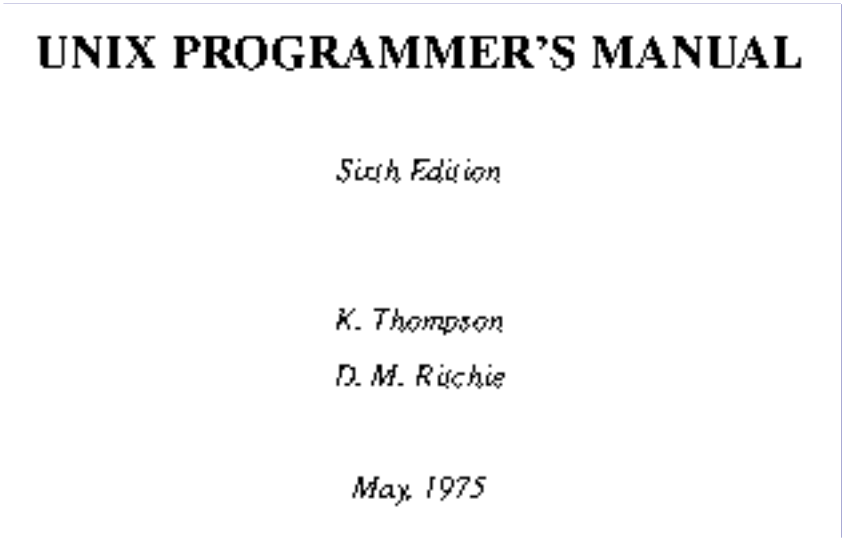
unix manuals – rcslog nroff

1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)
1971: roff (UNIX, PDP-11)
1973: nroff, troff (UNIX, PDP-11)
1975: nroff, troff (UNIX, C)


UNIX PROGRAMMER'S MANUAL

Sixth Edition

K. Thompson

D. M. Ritchie

May, 1975

1980

1990

2000

2010

2011: BSDCan

Ref: http://wwwlehre.dhbw-stuttgart.de/~helbig/os/v6/doc/index.html (UNIX Programmer's Manuals, Sixth Ed.)

This manual is divided into eight sections:

I.	Commands
II.	System calls
III.	Subroutines
IV.	Special files
V.	File formats and conventions
VI.	User-maintained programs
VII.	User-maintained subroutines
VIII.	Maintenance

unix manuals – rcslog ditroff

1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)
1971: roff (UNIX, PDP-11)
1973: nroff, troff (UNIX, PDP-11)
1975: nroff, troff (UNIX, C)

1980

1982: ditroff (UNIX, C)

1990

2000

2010

2011: BSDCan

Ref: http://wwwlehre.dhbw-stuttgart.de/~helbig/os/v6/doc/index.html (UNIX Programmer's Manuals, Sixth Ed.)

unix manuals – rcslog groff

1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)
1971: roff (UNIX, PDP-11)
1973: nroff, troff (UNIX, PDP-11)
1975: nroff, troff (UNIX, C)

1980

1982: ditroff (UNIX, C)

1990

1990: groff (GNU, C++)

2000

2010

2011: BSDCan

Ref: http://www.gnu.org/software/groff (GNU troff)

unix manuals – rcslog awf

This is awf, the Amazingly Workable Formatter -- a "nroff -man" or (subset) "nroff -ms" clone written entirely in (old) awk.

It is slow and has many restrictions, but does a decent job on most manual pages and simple -ms documents, and isn't subject to AT&T's brain-damaged licensing that denies many System V users any text formatter at all. It is also a text formatter that is simple enough to be tinkered with, for people who want to experiment.

Type "make r" to run a regression test, formatting the manual page (awf.1) and comparing it to a preformatted copy (awf.1.out). Type "make install" to install it. Pathnames may need changing.

I don't know whether awf will run on 16-bit machines. Data requirements are modest, but I fear the programs are probably big enough to run awk out of space.

I can't believe I really wrote this.

Henry Spencer at U of Toronto Zoology
henry@zoo.toronto.edu utzoo!henry
13 July 1990

1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)
1971: roff (UNIX, PDP-11)
1973: nroff, troff (UNIX, PDP-11)
1975: nroff, troff (UNIX, C)

1980

1982: ditroff (UNIX, C)

1990

1990: groff (GNU, C++), awf (awk)

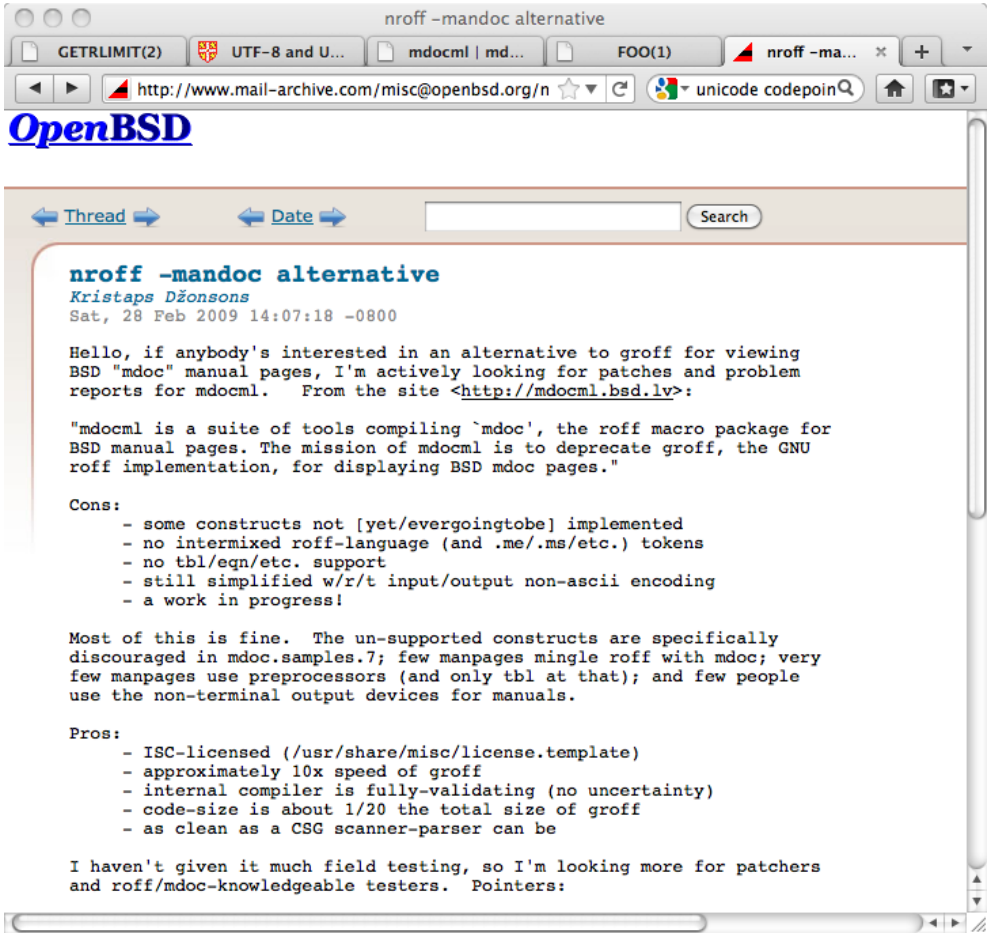
1995 (?): cawf (Minix?, C)

2000

2010

2011: BSDCan

unix manuals – rcslog mandoc



1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)
1971: roff (UNIX, PDP-11)
1973: nroff, troff (UNIX, PDP-11)
1975: nroff, troff (UNIX, C)

1980

1982: ditroff (UNIX, C)

1990

1990: groff (GNU, C++), awf (awk)

1995 (?): cawf (Minix, C)

2000

2008: mandoc (BSD.lv, C)

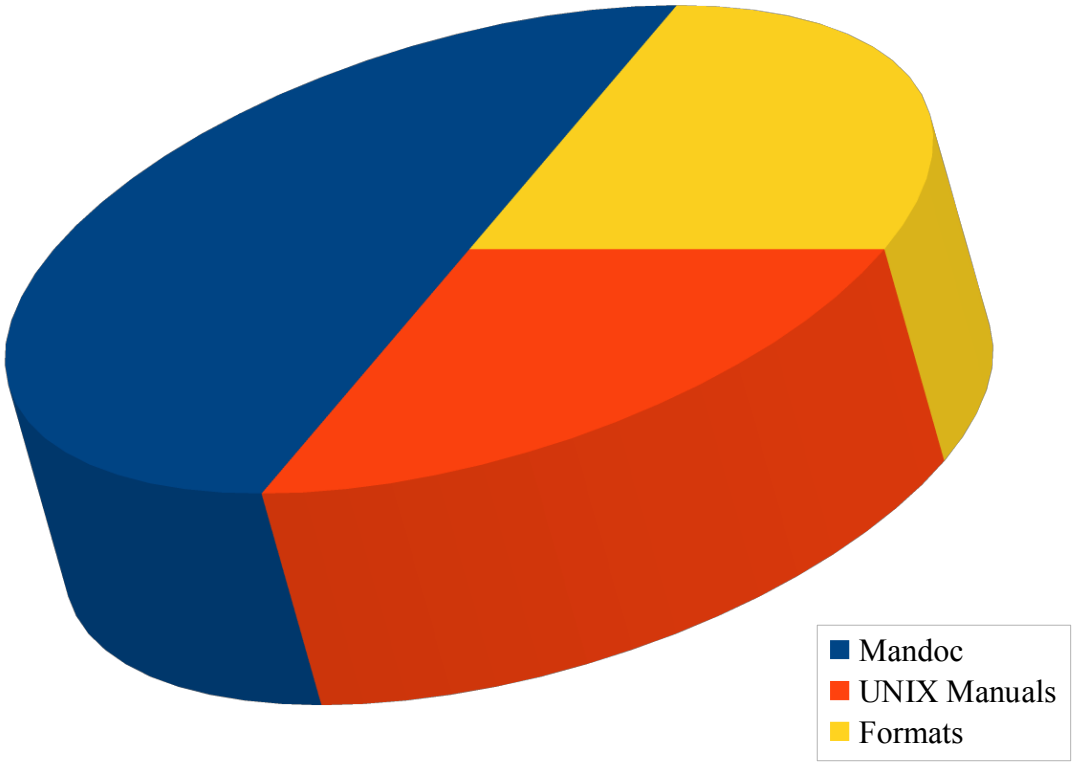
2010

2011: BSDCan

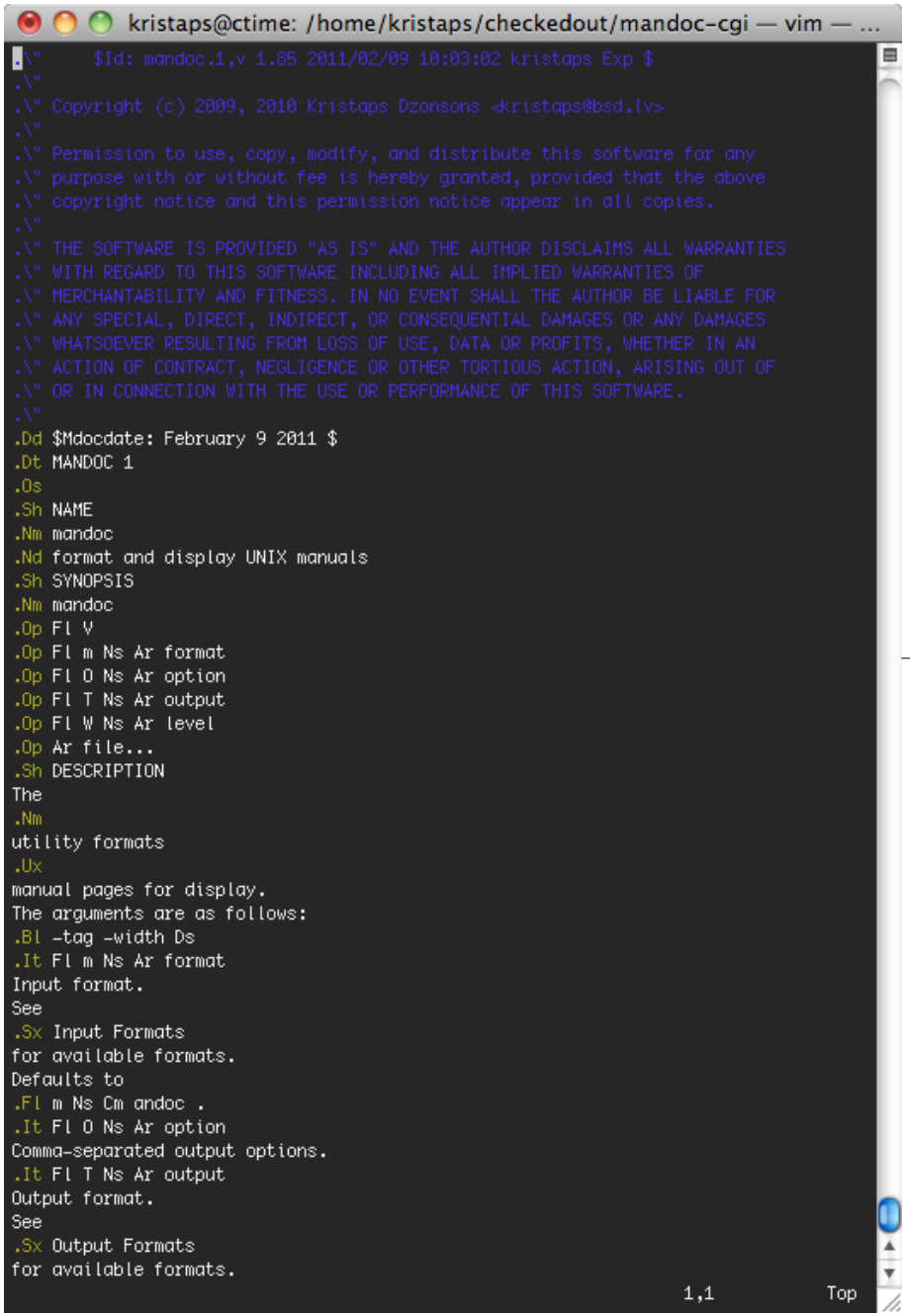
Ref: http://www.mdocml.bsd.lv (Mandoc)

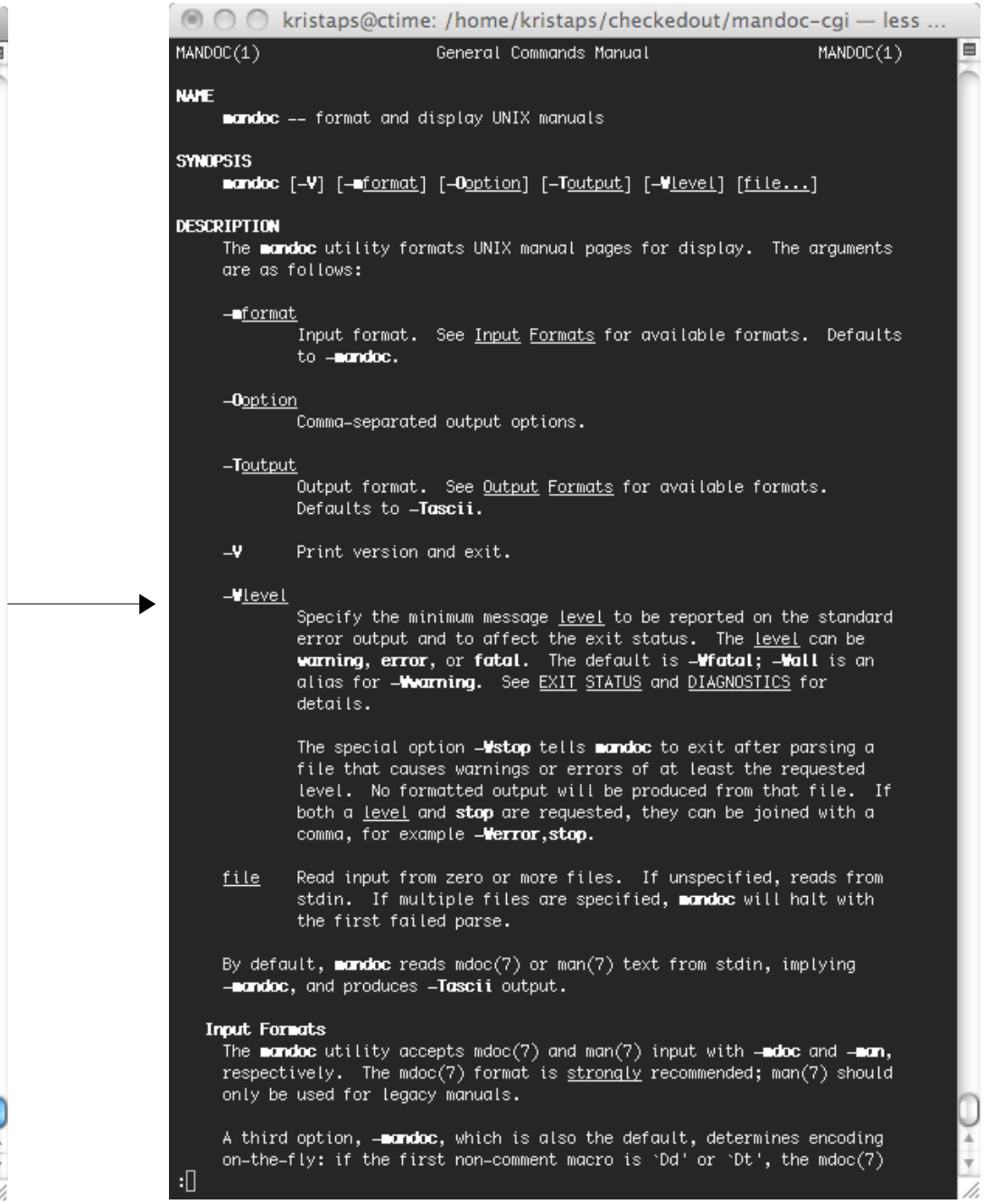
structure

- BSDCan-2011
  - UNIX Manuals
  - Formats
  - Mandoc
    http://mdocml.bsd.lv
    - mandoc.cgi
      http://mdocml.bsd.lv/mandoc-cgi



unix manuals – man





roff (Turing-complete, context-sensitive grammar)

- variables (typed-ish)
- looping
- conditionals
- functions

roff (Turing-complete, context-sensitive grammar)

- variables (typed-ish)
- looping
- conditionals
- functions

```
.ds stringvar anything
```

```
.ds hist (Historic groff only.)
```

```
.nr register +-N M
```

```
.nr nS 1
```

```
.if \\n(nS>0 \{\
.  if \\n(fZ>0 .br
.\}
```

roff (Turing-complete, context-sensitive grammar)

- variables (typed-ish)
- looping → .while cond *anything*
- conditionals
- functions

roff (Turing-complete, context-sensitive grammar)

- variables (typed-ish)
- looping
- conditionals
- functions

```
.if cond anything
```

```
.if \\n(nS>0 \{\
.  if \\n(fZ>0 .br
.\}
```

roff (Turing-complete, context-sensitive grammar)

- variables (typed-ish)
- looping
- conditionals
- functions
- macros

```
.de symbol anything
```

```
.de Ad
.if \\n(aC==0 \{\
.  ie \\n(.$==0 .tm Usage: .Ad address ... \\*(Pu (#\\n(.c)
.  el \{\
.          ds mN Ad
.          aV \\$1 \\$2 \\$3 \\$4 \\$5 \\$6 \\$7 \\$8 \\$9
.  \}
.\}
.if \\n(aC>\\n(aP \{\
.       as b1 \\*(aD
.       nr aP \\n(aP+1
.       nr cF \\n(.f
.       nr cZ \\n(.s
.       nR
.\}
..
```

unix manuals – rcslog macros

1960

1964: RUNOFF (CTSS)
1966: runoff (Multics)

1970

1970: runoff (UNIX, PDP-7, PDP-11)
1971: roff (UNIX, PDP-11), *me* (V1)
1973: nroff, troff (UNIX, PDP-11)
1975: nroff, troff (UNIX, C), *man* (V7)

1980

1982: ditroff (UNIX, C)

1990

1990: groff (GNU, C++), awf (awk)
1992: *mdoc* (4.4-BSD)

1995 (?): cawf (Minix, C)

2000

2008: mandoc (BSD.lv, C)

2010

2011: BSDCan

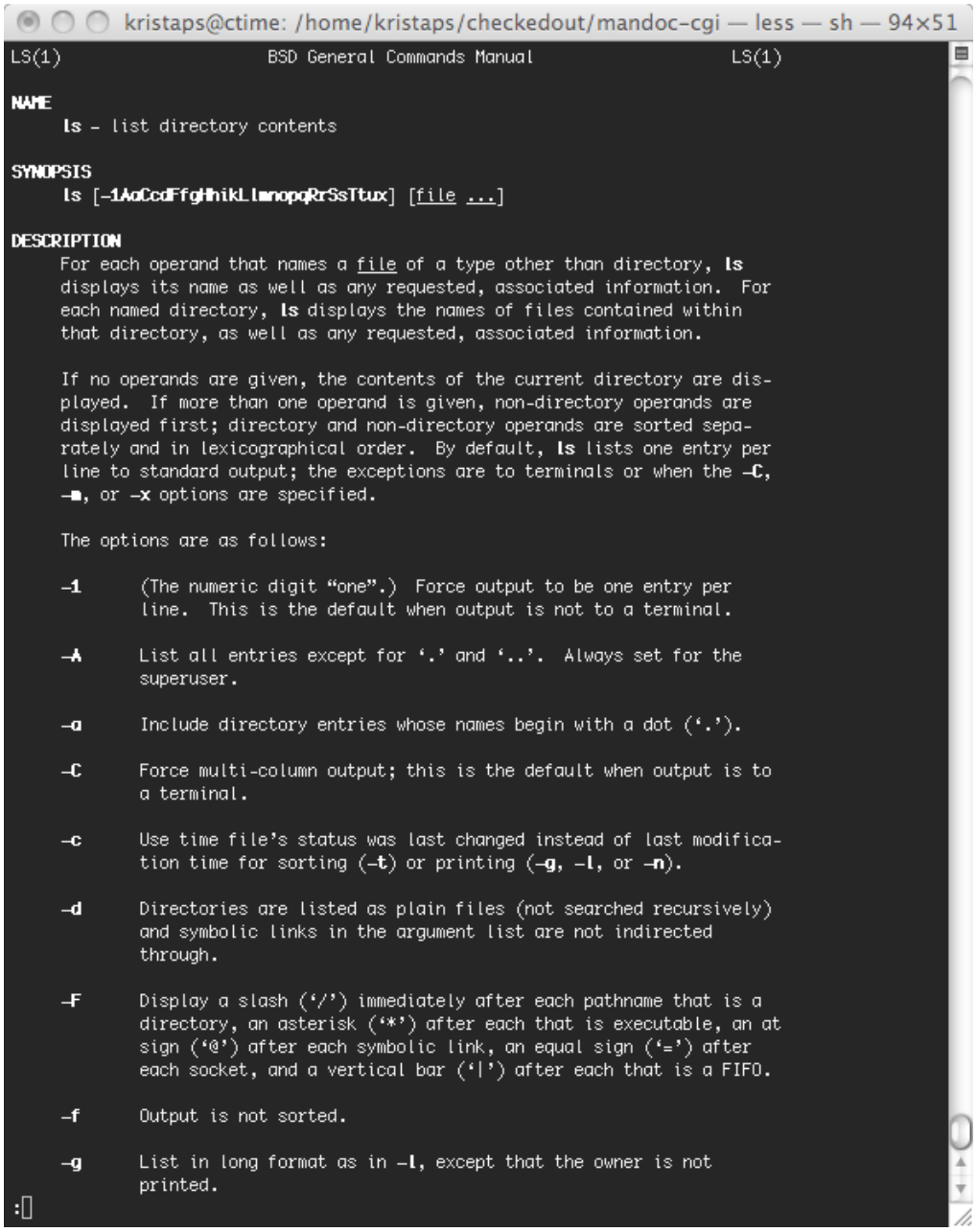
Ref: http://www.mdocml.bsd.lv (Mandoc)

unix manuals – *man*

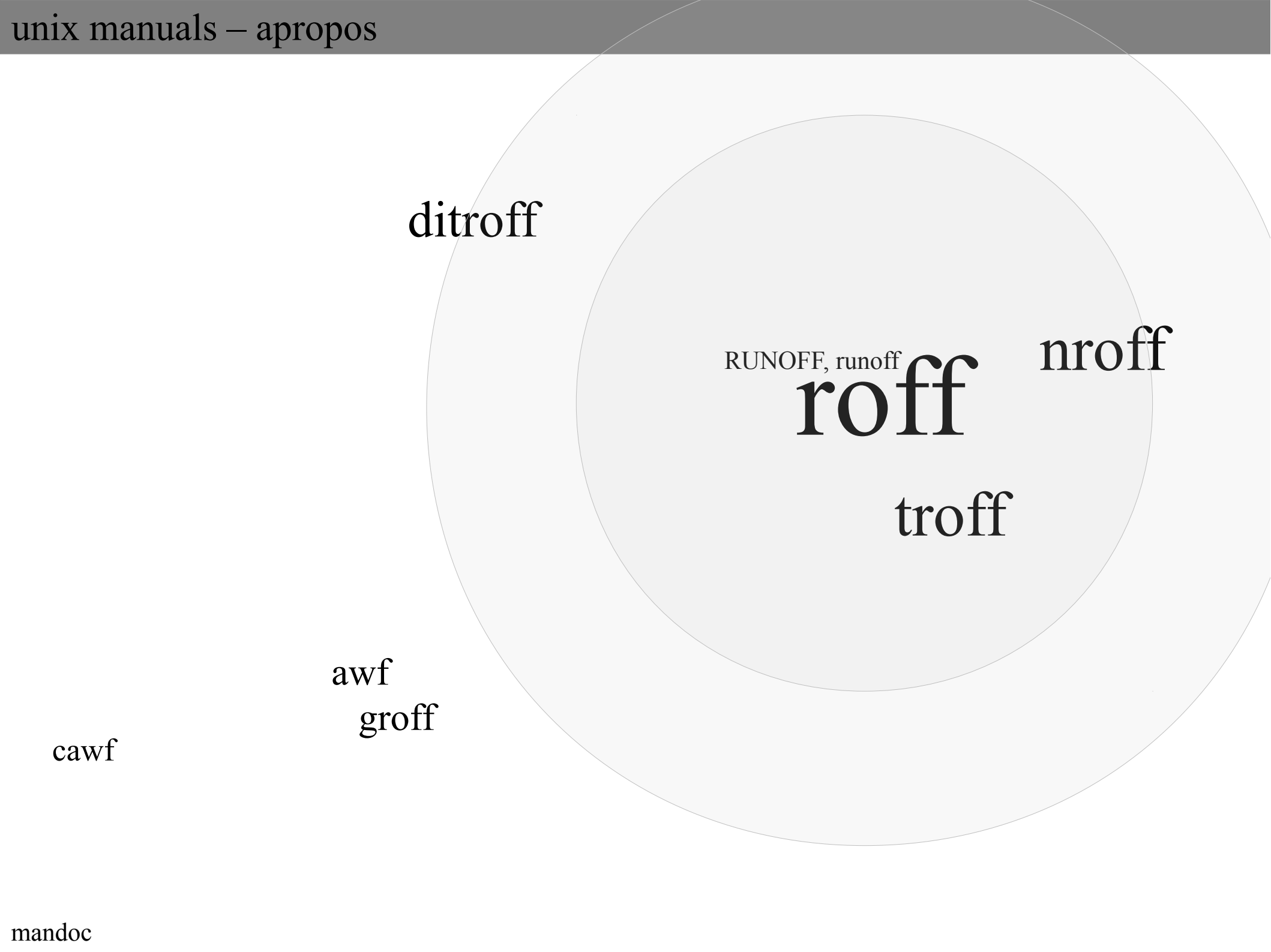
```
.TH GCC 1 "\*(Dt" "GNU Tools" "GNU Tools"
.SH NAME
gcc, g++ \- GNU project C and C++ Compiler (gcc-2.95)
.SH SYNOPSIS
.B gcc
.RI "[ " option " | " filename " ].\|.\|."
.br
.B g++
.RI "[ " option " | " filename " ].\|.\|."
.SH WARNING
The compiler bundled with OpenBSD differs significantly from the actual
2.95.3 release of the FSF. See gcc-local(1) for details.
.SH WARNING
The information in this man page is an extract from the full
documentation of the GNU C compiler, and is limited to the meaning of
the options.
.PP
This man page is not kept up to date except when volunteers want to
maintain it.  If you find a discrepancy between the man page and the
software, please check the Info file, which is the authoritative
documentation.
.PP
.PP
If we find that the things in this man page that are out of date cause
significant confusion or complaints, we will stop distributing the man
page.  The alternative, updating the man page when we update the Info
file, is impossible because the rest of the work of maintaining GNU CC
leaves us no time for that.  The GNU project regards man pages as
obsolete and should not let them take time away from other things.
.PP
For complete and current documentation, refer to the Info file `\|\c
.B gcc\c
\&\|' or the manual
```

unix manuals – *mdoc*

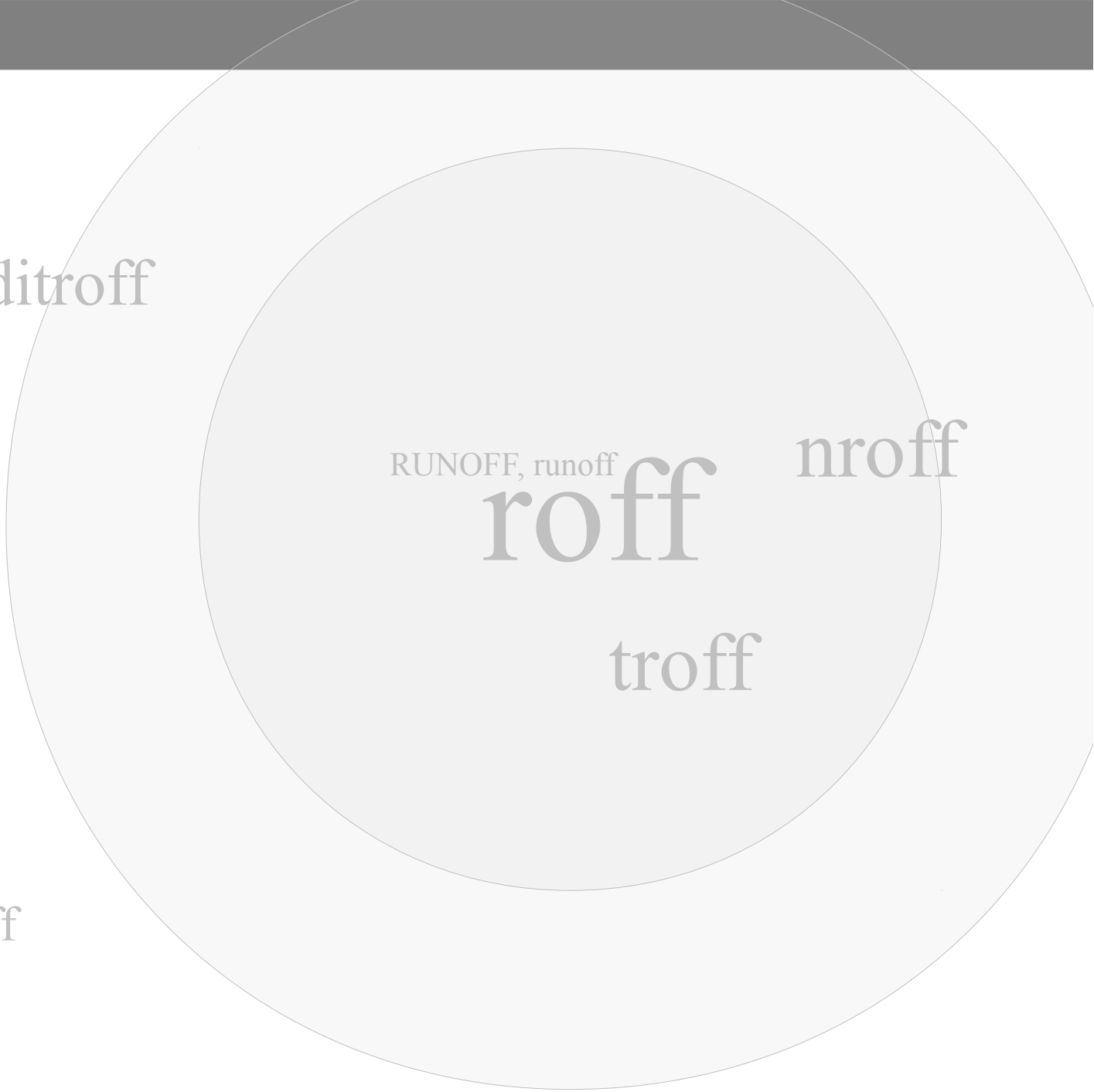
```
.Dd $Mdocdate: March 5 2011 $
.Dt LS 1
.Os
.Sh NAME
.Nm ls
.Nd list directory contents
.Sh SYNOPSIS
.Nm ls
.Op Fl 1AaCcdFfgHhikLlmnopqRrSsTtux
.Op Ar
.Sh DESCRIPTION
For each operand that names a
.Ar file
of a type other than directory,
.Nm
displays its name as well as any requested,
associated information.
For each named directory,
.Nm
displays the names of files contained
within that directory, as well as any requested, associated
information.
.Pp
If no operands are given, the contents of the current
directory are displayed.
If more than one operand is given,
non-directory operands are displayed first; directory
and non-directory operands are sorted separately and in
lexicographical order.
By default,
.Nm
lists one entry per line to standard
output; the exceptions are to terminals or when the
.Fl C ,
```



unix manuals – apropos



unix manuals – apropos



docbook, 1991
linuxbook, 1994

awf
groff
cawf

texinfo, 1993

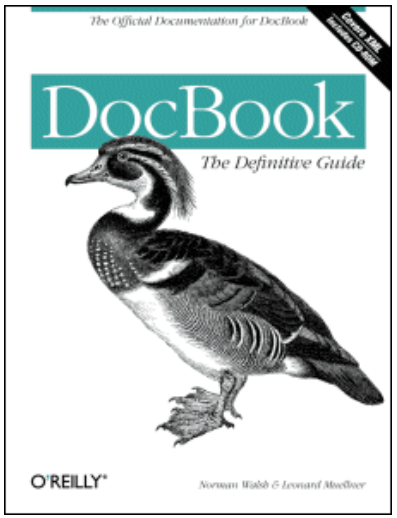
unix manuals – rcslog docbook

1990

1990: groff (GNU, C++)
1991: DocBook (O'Reilly, SGML)

1994: LinuxBook (Linux, SGML)

1995



1998: DocBook (OASIS, XML)
1998: Simplified DocBook

2000

2005

2010

2011: BSDCan

Ref: http://www.docbook.org/whatis (DocBook History)
Ref: http://www.w3.org/Tools/Linuxdoc-SGML.html (LinuxBook Announcement)
Ref: http://www.docbook.org/schemas/simplified (Simplified DocBook)

unix manuals – cat

```
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<refentry>
<refentryinfo>
        <date>2001-01-01</date>
</refentryinfo>
<refmeta>
        <refentrytitle>
                <application>foo</application>
        </refentrytitle>
        <manvolnum>1</manvolnum>
        <refmiscinfo>foo 1.0</refmiscinfo>
</refmeta>
<refnamediv>
        <refname>
                <application>foo</application>
        </refname>
        <refpurpose>
        Does nothing useful.
        </refpurpose>
</refnamediv>
<refsynopsisdiv>
        <refsynopsisdivinfo>
                <date>2001-01-01</date>
        </refsynopsisdivinfo>
        <cmdsynopsis>
        <command>foo</command>
<arg><option>-f </option><replaceable class="parameter">bar</replaceable></arg>
<arg><option>-d<replaceable class="parameter">n</replaceable></option></arg>
<arg rep="repeat"><replaceable class="parameter">file</replaceable></arg>
        </cmdsynopsis>
</refsynopsisdiv>
<refsect1>
        <refsect1info>
                <date>2001-01-01</date>
        </refsect1info>
        <title>DESCRIPTION</title>
        <para>
        <command>foo</command> does nothing useful.
        </para>
</refsect1>
<refsect1>
        <title>OPTIONS</title>
        <variablelist>
        <varlistentry>
                <term>-f <replaceable class="parameter">bar</replaceable></term>
                <listitem>
                        <para>
                        Takes <filename>bar</filename> as it's run
                        control file. If this were a real program,
                        there might be more to say here about what
                        bar is and how it will be used.
                        </para>
                </listitem>
        </varlistentry>
        <varlistentry>
                <term>-d<replaceable class="parameter">n</replaceable></term>
                <listitem>
                        <para>
                        Do something, where integer
                        <replaceable class="parameter">n</replaceable>
                        specifies how many times.
                        </para>
                </listitem>
        </varlistentry>
        <varlistentry>
                <term><replaceable class="parameter">file...</replaceable></term>
                <listitem>
                        <para>
                        Processes the files in the order listed,
                        sending all output to stdout.
                        </para>
                </listitem>
        </varlistentry>
        </variablelist>
</refsect1>
<refsect1>
        <title>USAGE</title>
        <para>
        <command>foo</command> -f foo.conf -d2 foodata.foo
        </para>
</refsect1>
<refsect1>
        <title>CAVEATS</title>
        <para>
        Other programs named <command>foo</command> may exist and actually
        do something!
        </para>
</refsect1>
<refsect1>
        <title>BUGS</title>
        <para>
        None.  Program does nothing.
        </para>
</refsect1>
<refsect1>
        <title>AUTHOR</title>
        <para>
        <author>
                <firstname>Foo</firstname>
                <othername role="mi">E</othername>
                <surname>Bar</surname>
                <contrib>Original author</contrib>
        </author>
        </para>
</refsect1>
</refentry>
```

unix manuals – cat

```
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<refentry>
<refentryinfo>
        <date>2001-01-01</date>
</refentryinfo>
<refmeta>
        <refentrytitle>
                <application>foo</application>
        </refentrytitle>
        <manvolnum>1</manvolnum>
        <refmiscinfo>foo 1.0</refmiscinfo>
</refmeta>
<refnamediv>
        <refname>
                <application>foo</application>
        </refname>
        <refpurpose>
        Does nothing useful.
        </refpurpose>
</refnamediv>
<refsynopsisdiv>
        <refsynopsisdivinfo>
                <date>2001-01-01</date>
        </refsynopsisdivinfo>
        <cmdsynopsis>
        <command>foo</command>
<arg><option>-f </option><replaceable class="parameter">bar</replaceable></arg>
<arg><option>-d<replaceable class="parameter">n</replaceable></option></arg>
<arg rep="repeat"><replaceable class="parameter">file</replaceable></arg>
        </cmdsynopsis>
</refsynopsisdiv>
<refsect1>
        <refsect1info>
                <date>2001-01-01</date>
        </refsect1info>
        <title>DESCRIPTION</title>
        <para>
        <command>foo</command> does nothing useful.
        </para>
```

```
FOO(1)                                                    FOO(1)


NAME
       foo - Does nothing useful.

SYNOPSIS
       foo [ -f bar ]  [ -dn ]  [ file... ]

DESCRIPTION
       foo does nothing useful.

OPTIONS
       -f bar Takes  bar as its run control file. If this were a
              real program, there might be more to say here about
              what bar is and how it will be used.

       -dn    Do  something,  where  integer n specifies how many
              times.

       file...
              Processes the files in the  order  listed,  sending
              all output to stdout.

USAGE
       foo -f foo.conf -d2 foodata.foo

CAVEATS
       Other  programs  named foo may exist and actually do some-
       thing!

BUGS
       None. Program does nothing.

AUTHOR
       Foo E Bar (Original author)
```

unix manuals – rcslog texinfo

1990

1990: groff (GNU, C++)
1991: DocBook (O'Reilly, SGML)

1993: Texinfo (GNU, info)
1994: LinuxBook (Linux, SGML)

1995

1998: DocBook (OASIS, XML)
1998: Simplified DocBook

2000

2005

2010

2011: BSDCan

Ref: http://www.gnu.org/software/texinfo/ (GNU Texinfo)

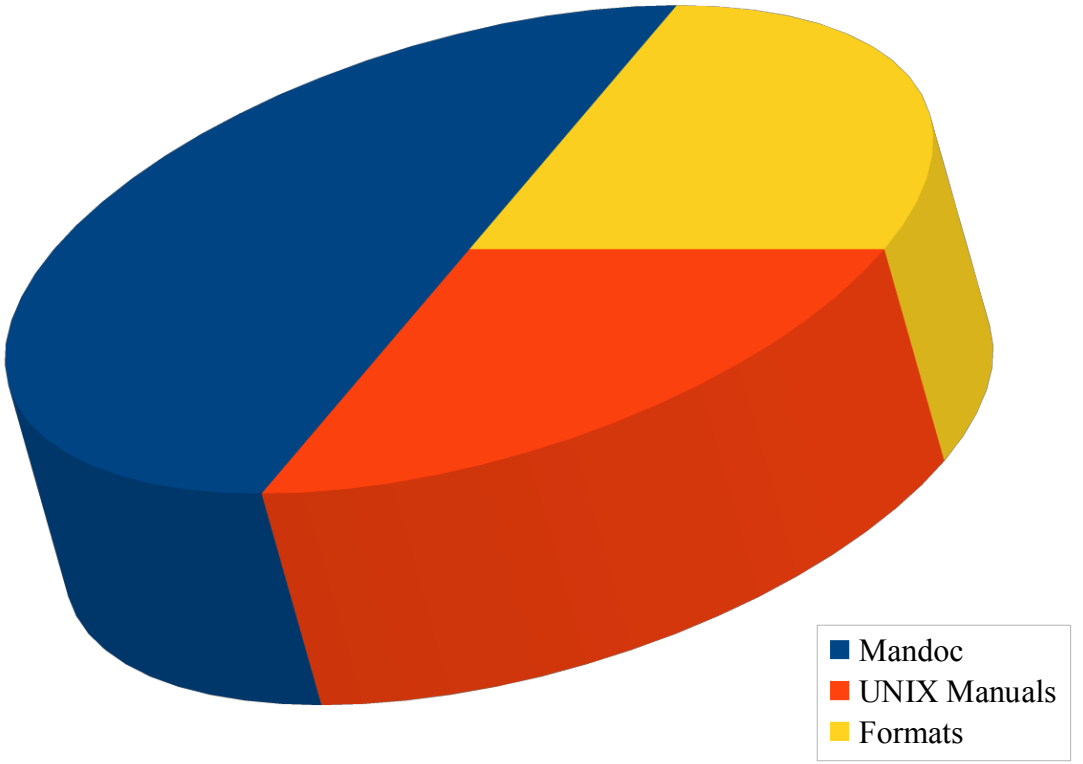
```
\input texinfo   @c -*-texinfo-*-
@comment $Id: GNU-Sample-Texts.html,v 1.7 2008/09/19 15:44:59 karl Exp $
@comment %**start of header
@setfilename sample.info
@include version.texi
@settitle GNU Sample @value{VERSION}
@syncodeindex pg cp
@comment %**end of header
@copying
This manual is for GNU Sample (version @value{VERSION}, @value{UPDATED}),
which is an example in the Texinfo documentation.

Copyright @copyright{} 2007 Free Software Foundation, Inc.

@quotation
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2 or
any later version published by the Free Software Foundation; with no
Invariant Sections, with the Front-Cover Texts being ``A GNU Manual,''
and with the Back-Cover Texts as in (a) below.  A copy of the
license is included in the section entitled ``GNU Free Documentation
License.''

(a) The FSF's Back-Cover Text is: ``You have the freedom to
copy and modify this GNU manual.  Buying copies from the FSF
supports it in developing GNU and promoting software freedom.''
@end quotation
@end copying

@dircategory Texinfo documentation system
@direntry
* sample: (sample)Invoking sample.
@end direntry

@titlepage
@title GNU Sample
@subtitle for version @value{VERSION}, @value{UPDATED}
@author A.U. Thor (@email{bug-texinfo@@gnu.org})
@page
@vskip 0pt plus 1filll
@insertcopying
@end titlepage

@contents

@ifnottex
@node Top
@top GNU Sample

This manual is for GNU Sample (version @value{VERSION}, @value{UPDATED}).
@end ifnottex

@menu
* Invoking sample::
* Copying This Manual::
* Index::
@end menu


@node Invoking sample
@chapter Invoking sample

@pindex sample
@cindex invoking @command{sample}

This is a sample manual.  There is no sample program to
invoke, but if there was, you could see its basic usage
and command line options here.


@node GNU Free Documentation License
@appendix GNU Free Documentation License

@include fdl.texi


@node Index
@unnumbered Index

@printindex cp

@bye
```

```
@contents

@ifnottex
@node Top
@top GNU Sample

This manual is for GNU Sample (version @value{VERSION}, @value{UPDATED}).
@end ifnottex

@menu
* Invoking sample::
* Copying This Manual::
* Index::
@end menu


@node Invoking sample
@chapter Invoking sample

@pindex sample
@cindex invoking @command{sample}

This is a sample manual.  There is no sample program to
invoke, but if there was, you could see its basic usage
and command line options here.


@node GNU Free Documentation License
@appendix GNU Free Documentation License

@include fdl.texi


@node Index
@unnumbered Index

@printindex cp

@bye
```

structure

BSDCan-2011

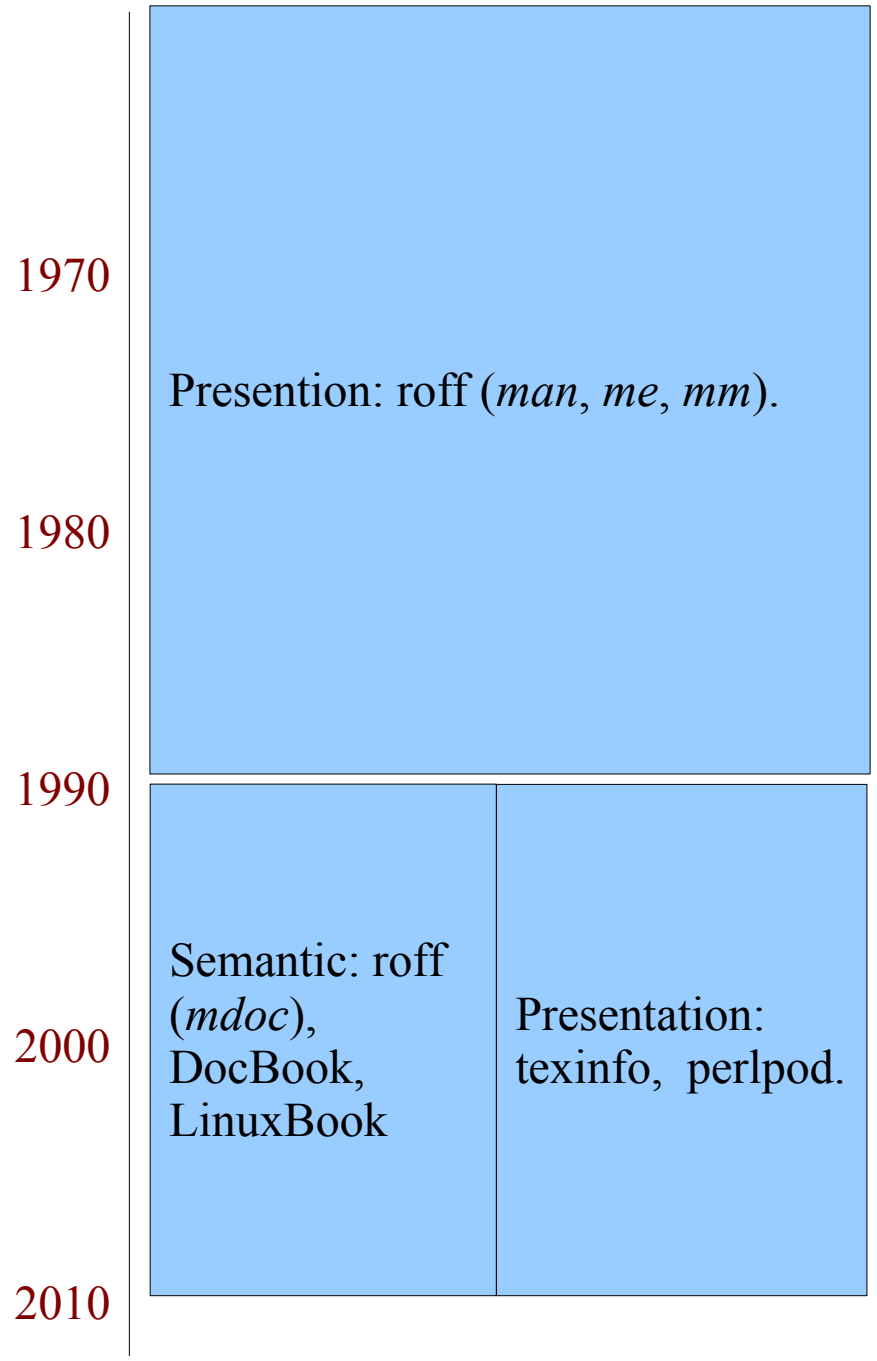
- UNIX Manuals
- Formats
- Mandoc
  http://mdocml.bsd.lv
  - mandoc.cgi
    http://mdocml.bsd.lv/mandoc-cgi



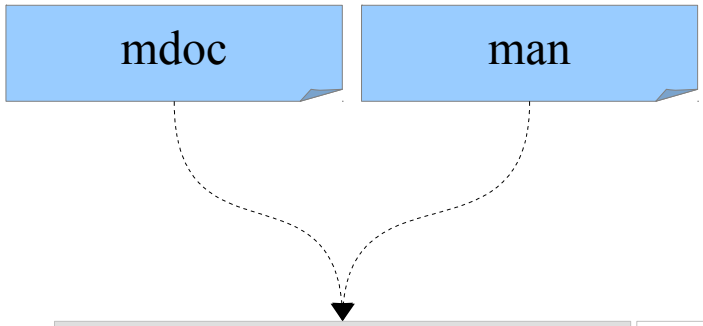
unix manuals – file

Encodes information regarding the *presentation style* of the document.

Encodes information regarding the *content* of the document.



unix manuals – file



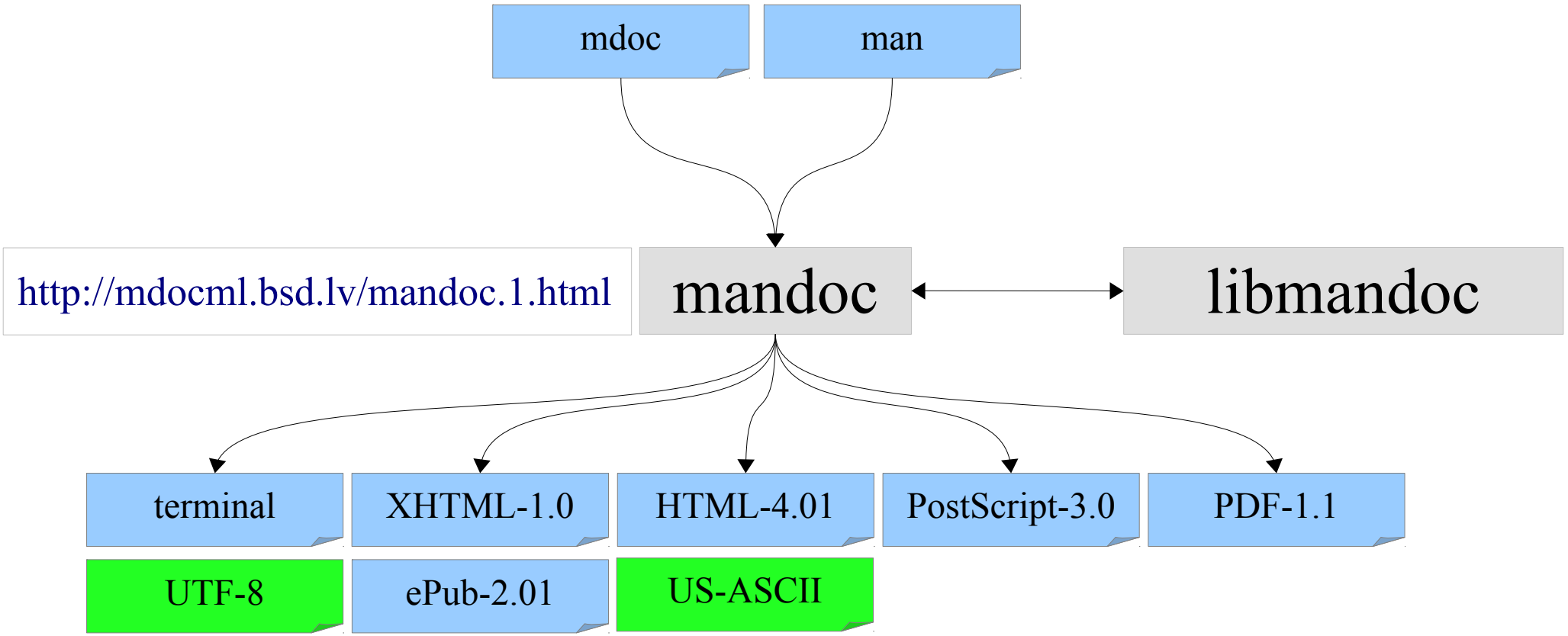
libmandoc

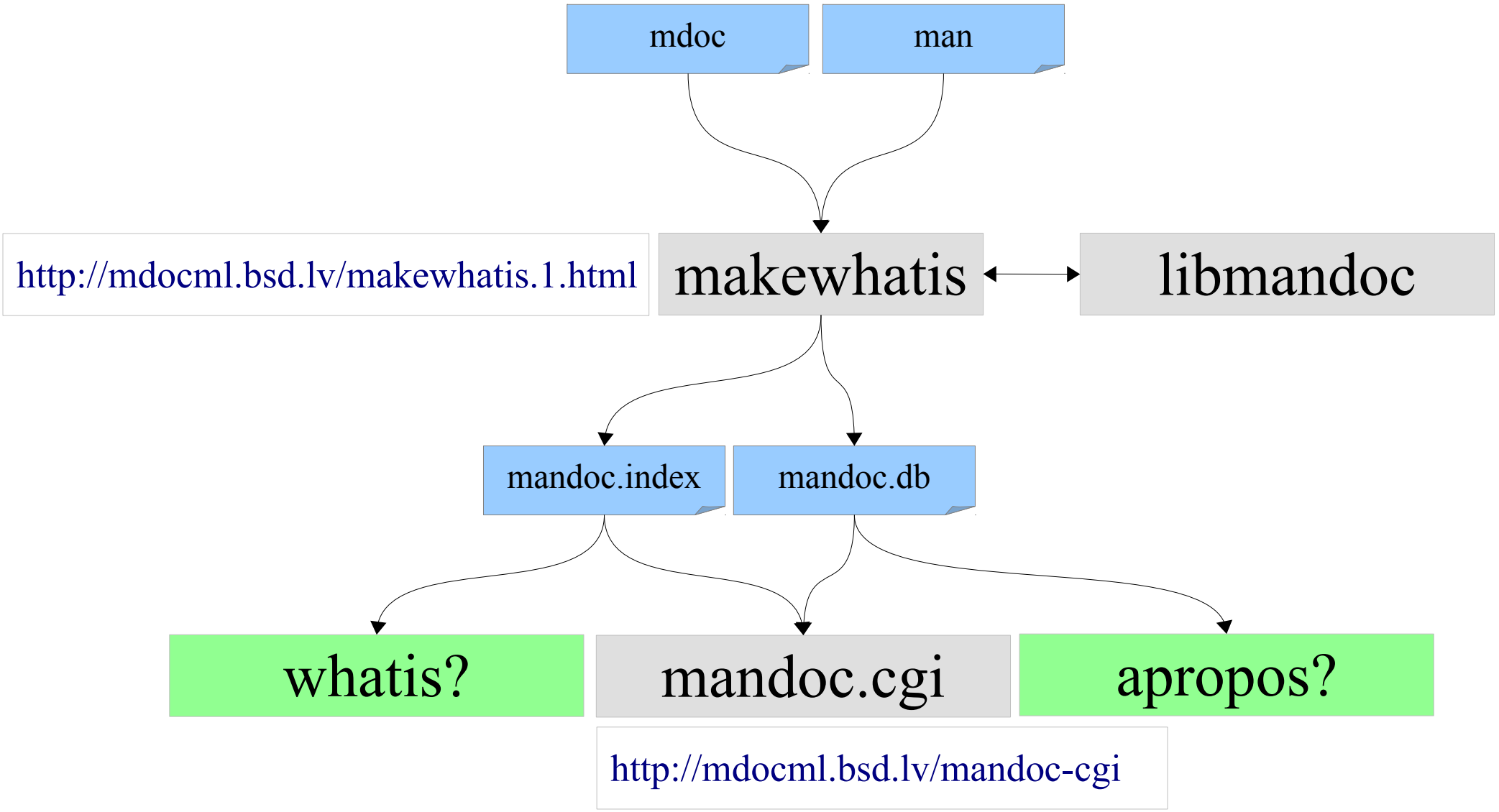
http://mdocml.bsd.lv/mandoc.3.html

semantic tree

```
Sh (block) 23:1
        Sh (block-head) 23:1
                SYNOPSIS (text) 23:4
        Sh (block-body) 23:1
                Nm (block) 24:1
                        Nm (block-head) 24:1
                                mandoc (text) 24:4
                        Nm (block-body) 24:1
                                Op (block) 25:1
                                        Op (block-head) 25:1
                                        Op (block-body) 25:1
                                                Fl (elem) 25:4
                                                        V (text) 25:7
```

unix manuals – file

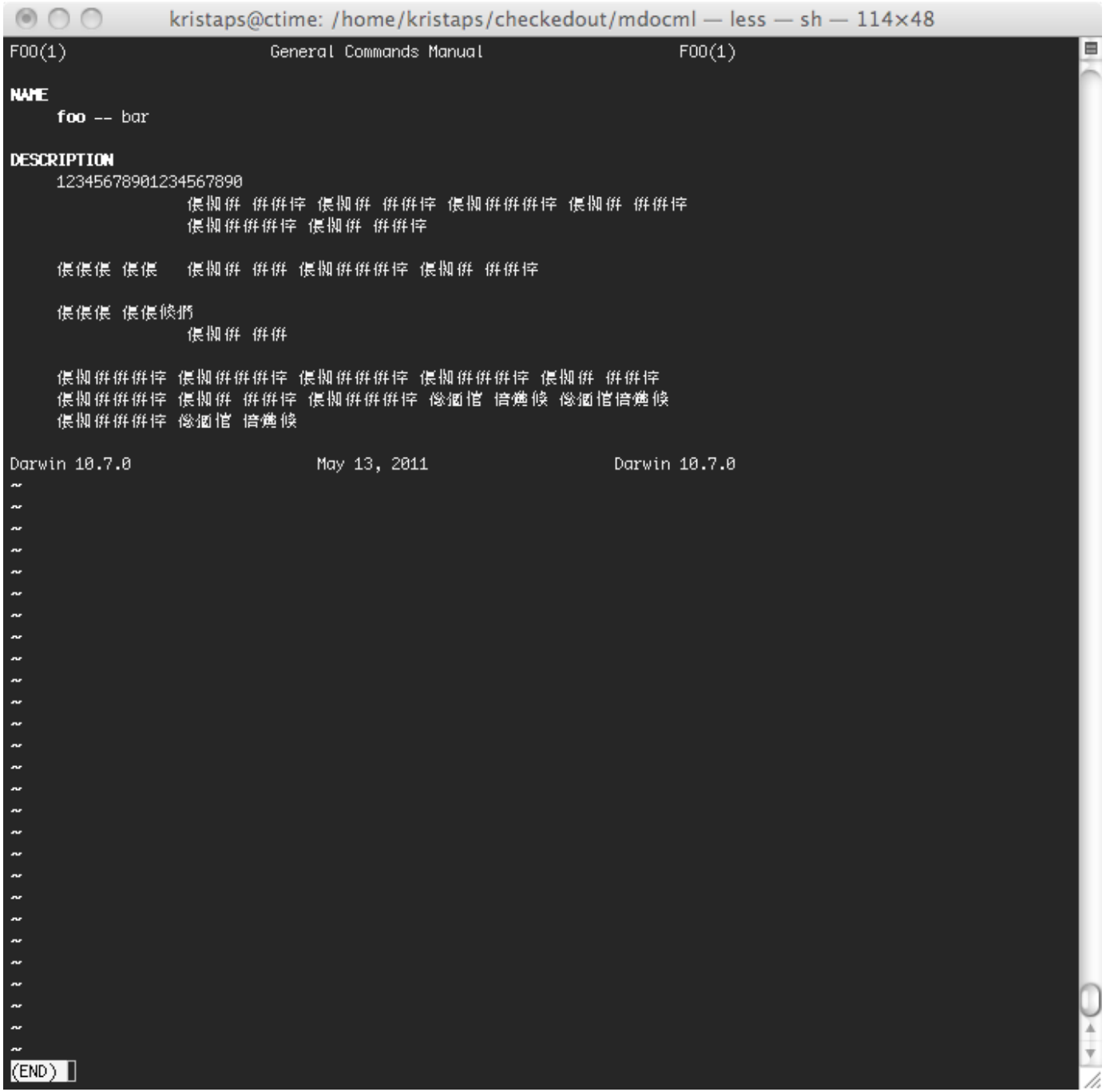



mdoc
man
http://mdocml.bsd.lv/makewhatis.1.html
makewhatis
libmandoc
mandoc.index
mandoc.db
whatis?
mandoc.cgi
apropos?
http://mdocml.bsd.lv/mandoc-cgi

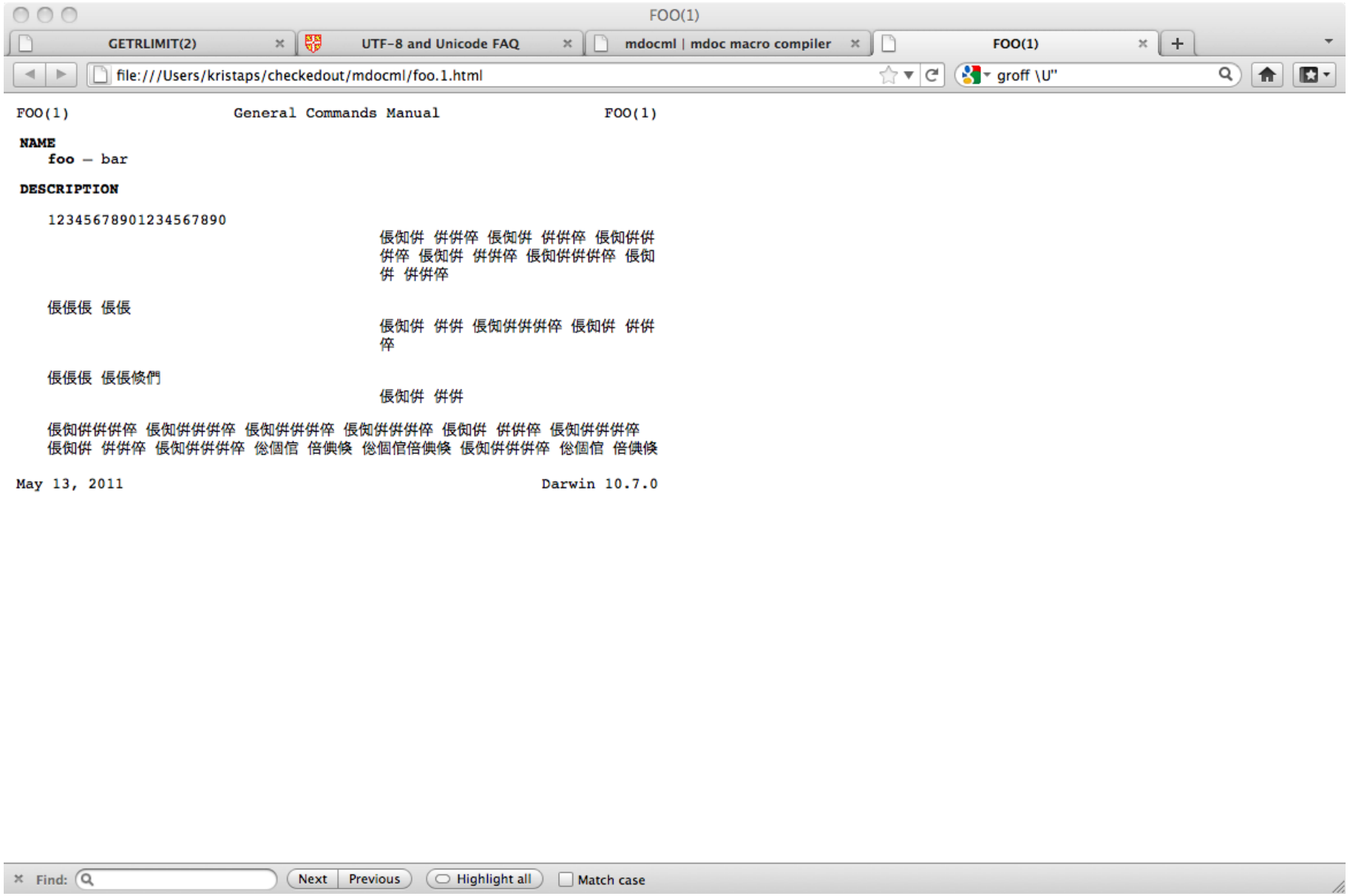
goodies: Makefile

```
.SUFFIXES: .1 .1.html .in.1 .1.xhtml .1.pdf

all: tool.1 tool.1.html

clean:
	rm -f tool.1 tool.1.html tool.1.xhtml tool.1.pdf

.in.1.1:
	mandoc -Tlint -Wstop $<
	cp $< $@

.1.1.xhtml:
	mandoc -Txhtml -Ostyle=style.css $< >$@

.1.1.html:
	mandoc -Thtml -Ostyle=style.css $< >$@

.1.1.pdf:
	mandoc -Tpdf $< >$@
```

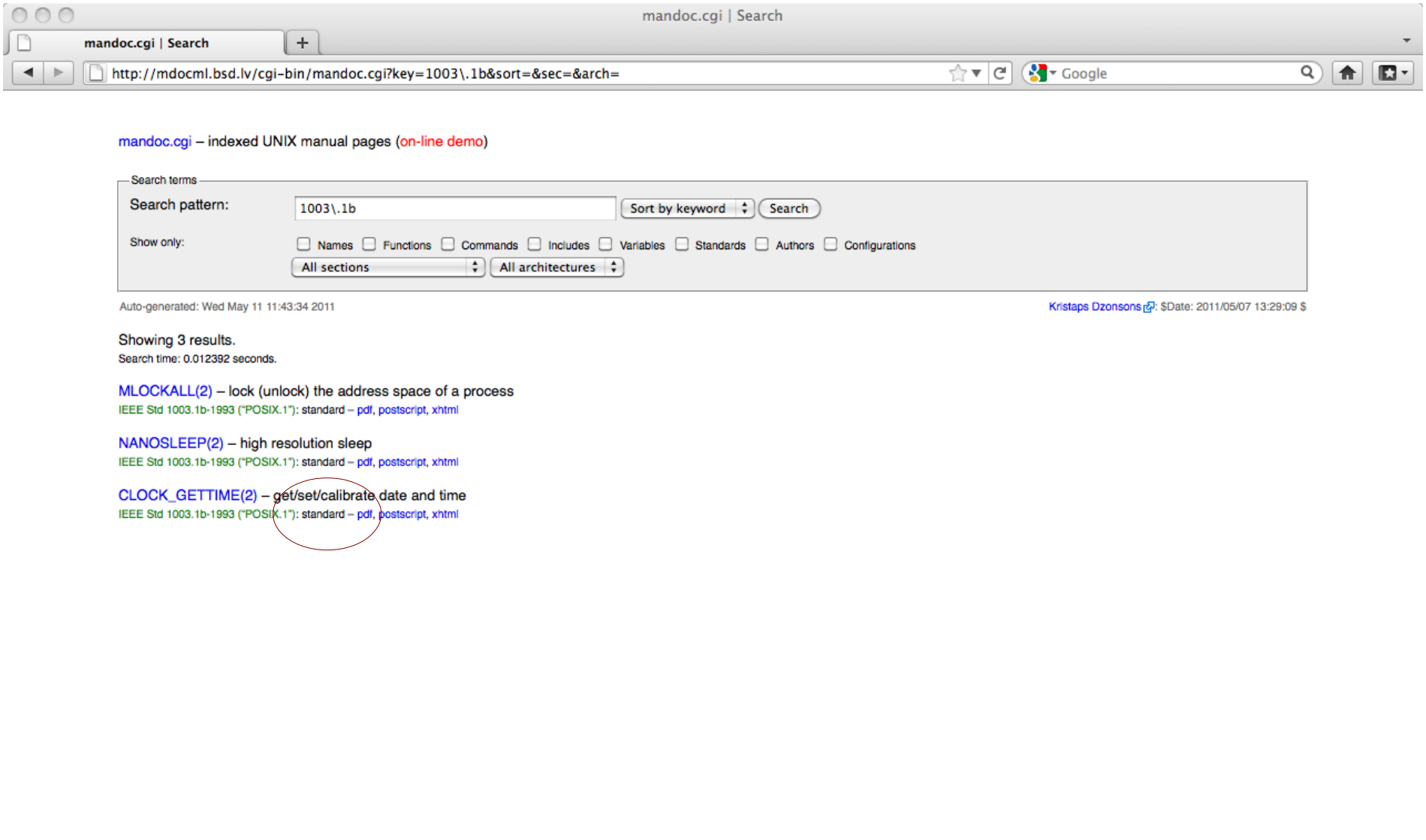
goodies: mandoc -Tutf8



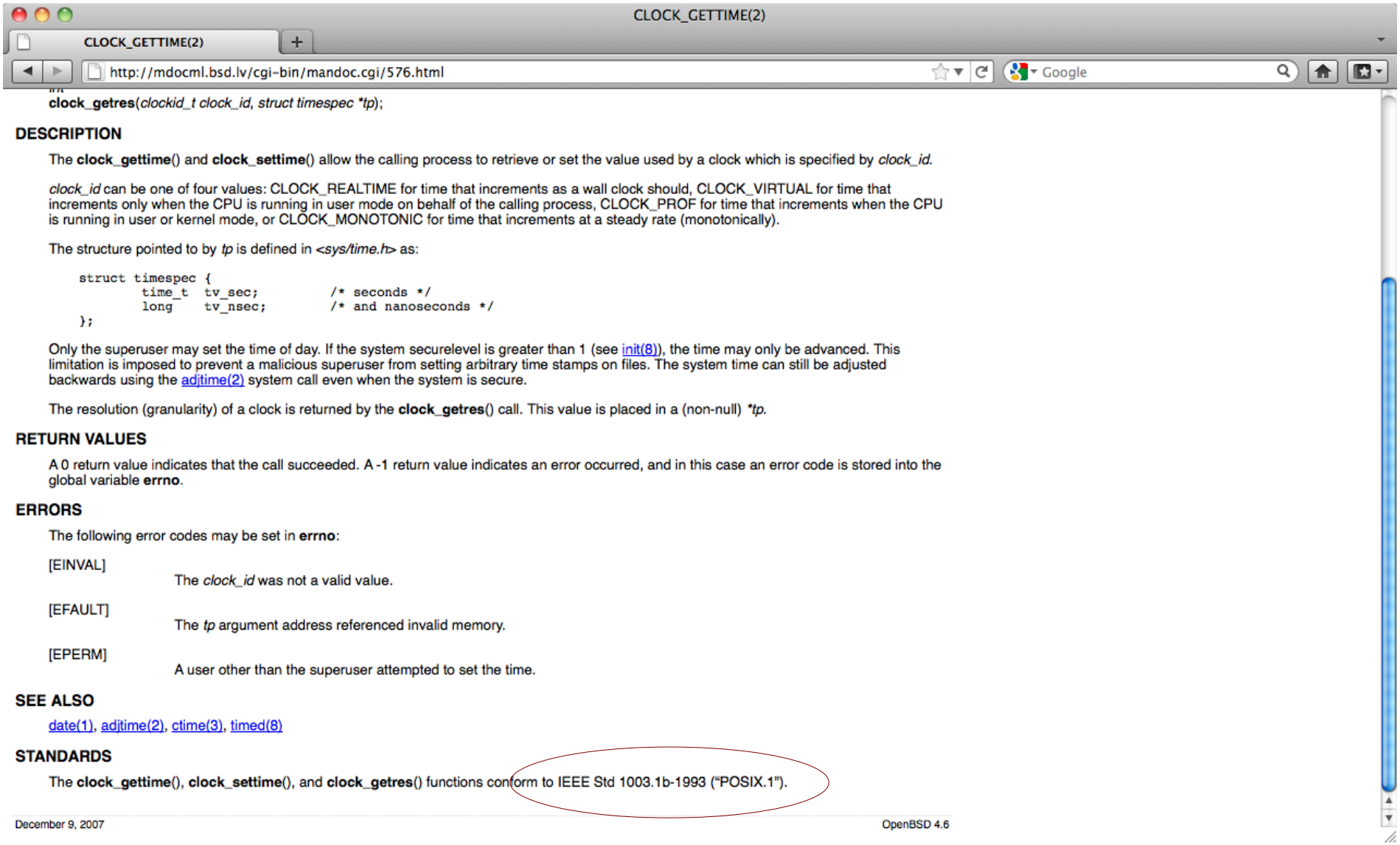
goodies: mandoc -Tutf8



goodies: mandoc.cgi



goodies: mandoc.cgi



clock_getres(*clockid_t clock_id*, *struct timespec *tp*);

DESCRIPTION

The **clock_gettime**() and **clock_settime**() allow the calling process to retrieve or set the value used by a clock which is specified by *clock_id*.

clock_id can be one of four values: CLOCK_REALTIME for time that increments as a wall clock should, CLOCK_VIRTUAL for time that increments only when the CPU is running in user mode on behalf of the calling process, CLOCK_PROF for time that increments when the CPU is running in user or kernel mode, or CLOCK_MONOTONIC for time that increments at a steady rate (monotonically).

The structure pointed to by *tp* is defined in <*sys/time.h*> as:

```
struct timespec {
        time_t  tv_sec;         /* seconds */
        long    tv_nsec;        /* and nanoseconds */
};
```

Only the superuser may set the time of day. If the system securelevel is greater than 1 (see init(8)), the time may only be advanced. This limitation is imposed to prevent a malicious superuser from setting arbitrary time stamps on files. The system time can still be adjusted backwards using the adjtime(2) system call even when the system is secure.

The resolution (granularity) of a clock is returned by the **clock_getres**() call. This value is placed in a (non-null) **tp*.

RETURN VALUES

A 0 return value indicates that the call succeeded. A -1 return value indicates an error occurred, and in this case an error code is stored into the global variable **errno**.

ERRORS

The following error codes may be set in **errno**:

[EINVAL]
The *clock_id* was not a valid value.

[EFAULT]
The *tp* argument address referenced invalid memory.

[EPERM]
A user other than the superuser attempted to set the time.

SEE ALSO

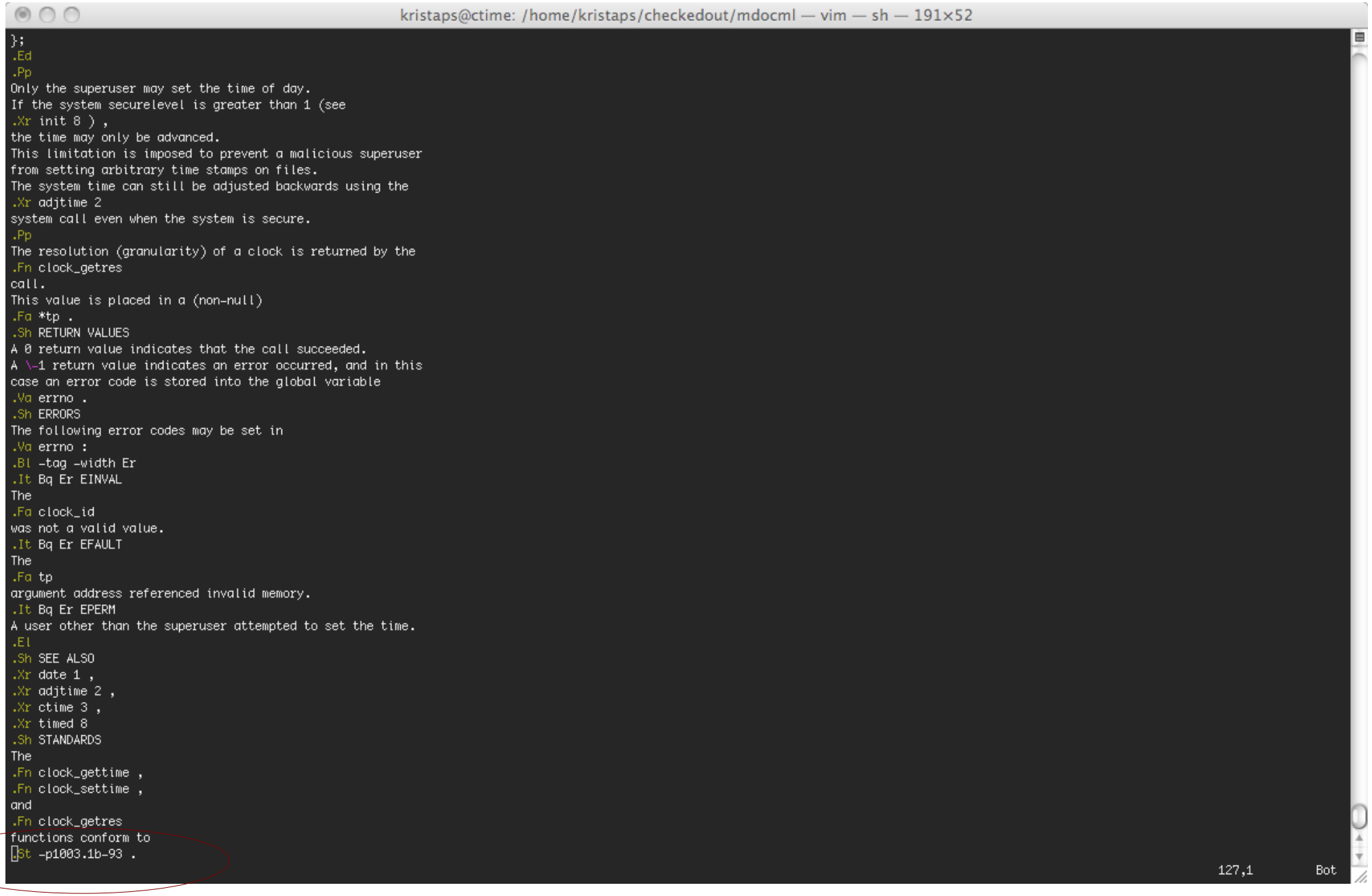
date(1), adjtime(2), ctime(3), timed(8)

STANDARDS

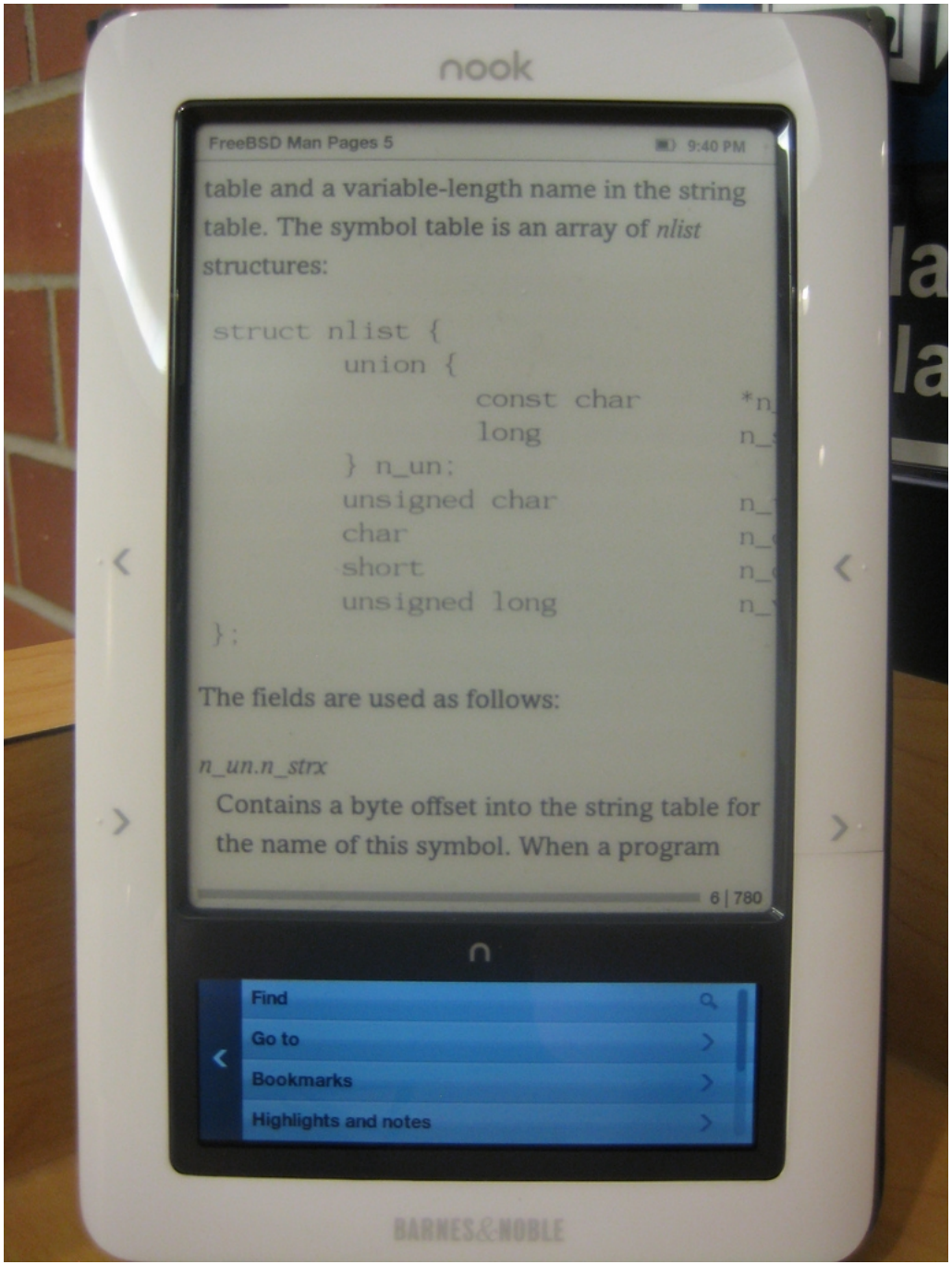
The **clock_gettime**(), **clock_settime**(), and **clock_getres**() functions conform to IEEE Std 1003.1b-1993 ("POSIX.1").

December 9, 2007 OpenBSD 4.6

goodies: mandoc.cgi

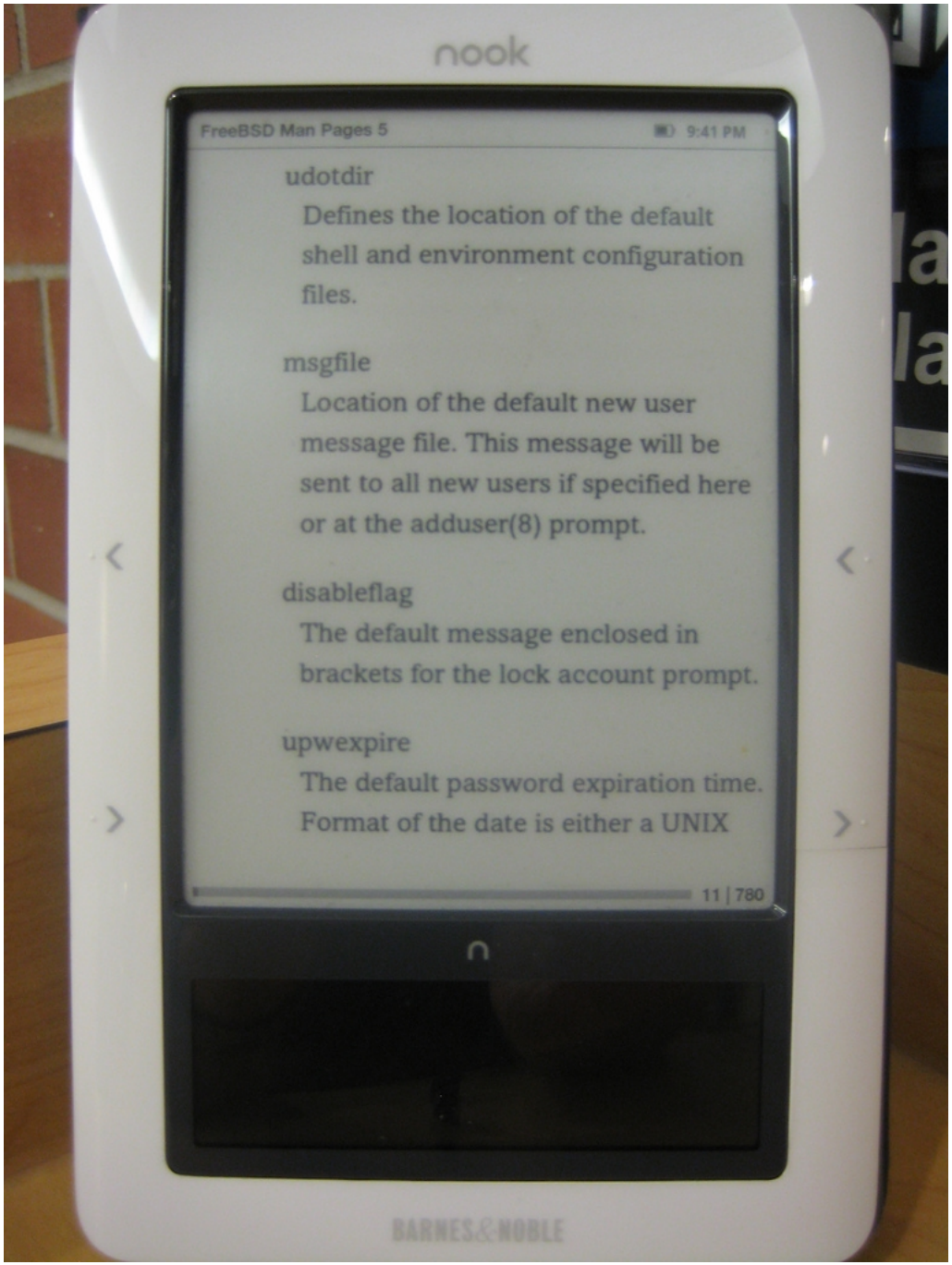


goodies: mandoc -Txhtml (epub)



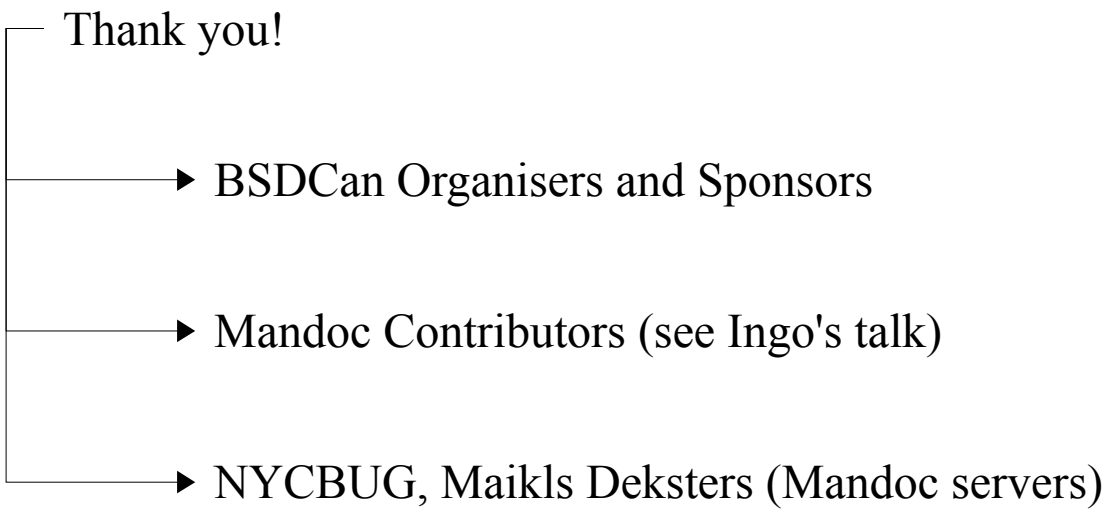
Ref: http://www.flickr.com/photos/bitgeist/5712000998 (Will Backman)

goodies: mandoc -Txhtml (epub)

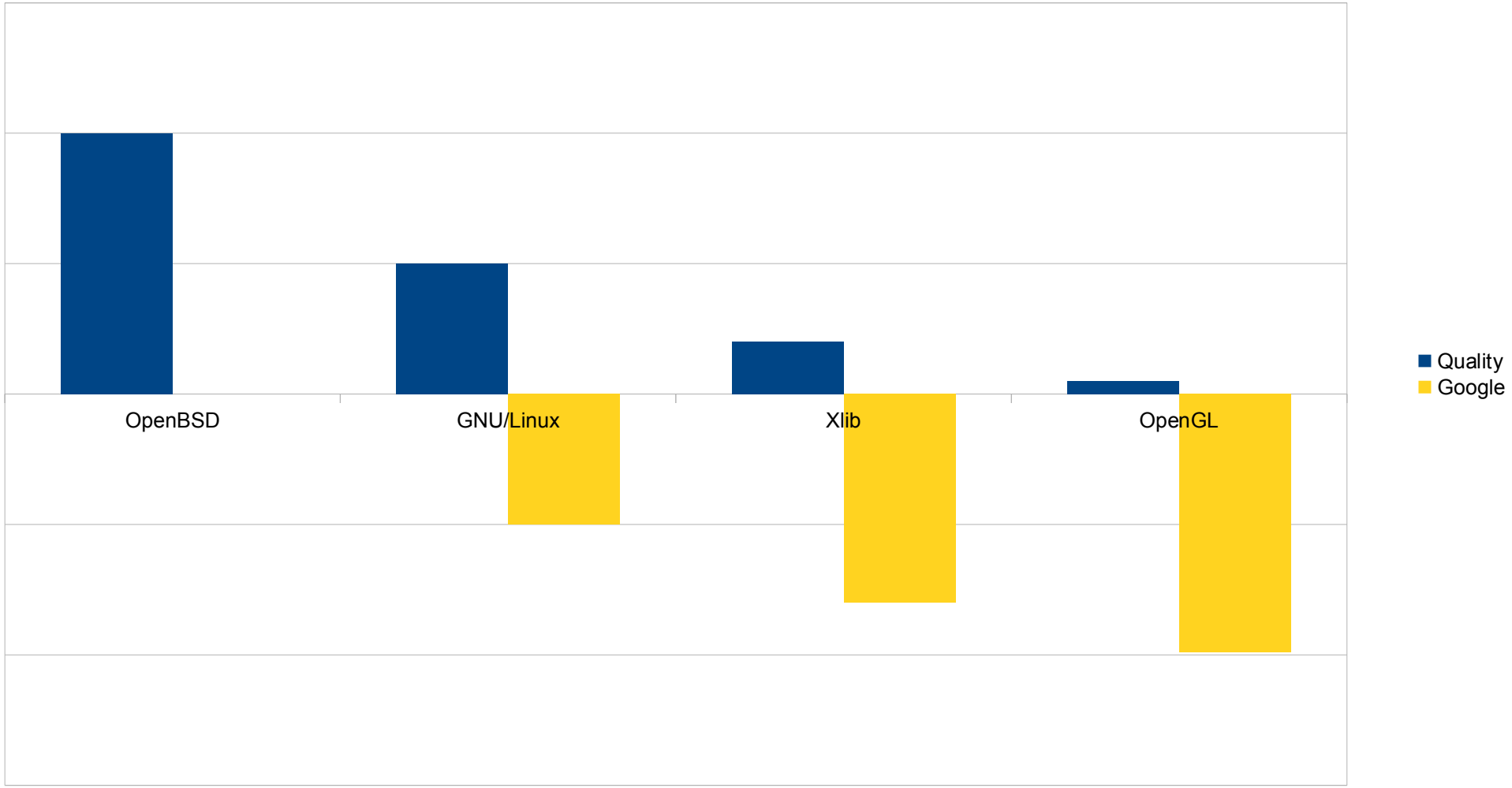


Ref: http://www.flickr.com/photos/bitgeist/5711441679 (Will Backman)

mandoc – thank you

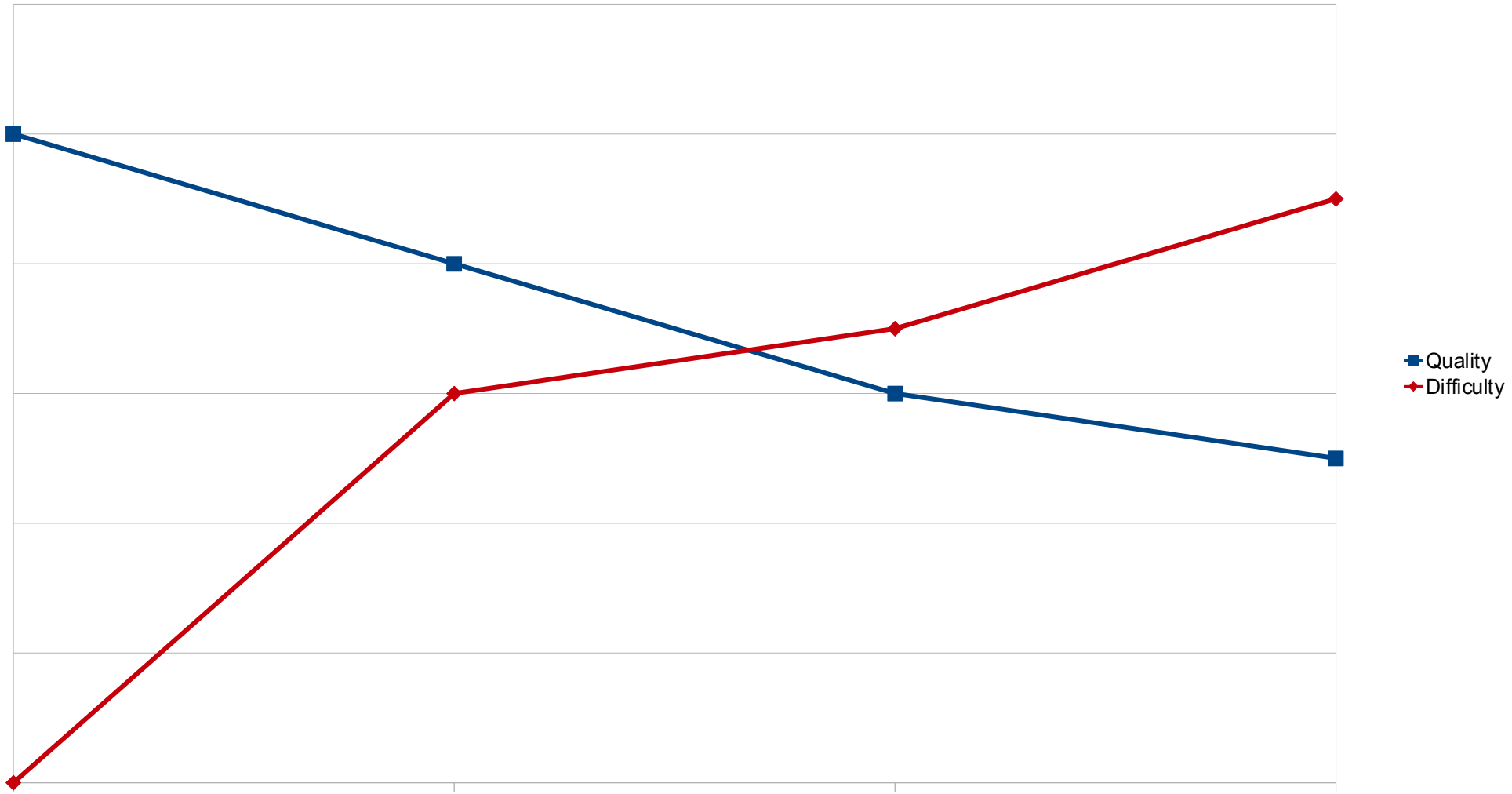


mandoc – more



Ref: Me.

mandoc – more



Ref: Me.

mandoc – more

?=