

CONTENTS

Fred Fish

COLLABORATORS

	<i>TITLE :</i> <i>CONTENTS</i>		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Fred Fish	September 19, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	CONTENTS	1
1.1	GNU and Other Utilities	1
1.2	Index of subdirectory: src	2
1.3	C++ class library for AmigaDOS functions	5
1.4	GNU automatic configuration generator.	5
1.5	GNU arbitrary precision calculator lang.	6
1.6	GNU binary file utilities.	6
1.7	GNU binary utilities, Amiga source	6
1.8	GNU parser generator yacc replacement	6
1.9	Compute & use CRC lists to verify files	7
1.10	Advanced desk calculator in EMACS elisp.	7
1.11	GNU utility to copy to/from archives.	8
1.12	Debugging malloc/realloc/free routines.	9
1.13	Macro based debugging library and tools.	10
1.14	GNU reverse-polish (RPN) desk calculator	10
1.15	GNU diff, diff3, sdiff and cmp utilities	11
1.16	Check DOS/SYSV filename limits.	11
1.17	8-bit-clean POSIX compliant line editor.	12
1.18	GNU Emacs editor	12
1.19	GNU Emacs editor	13
1.20	Fortran 77 to C translator	14
1.21	File management utilities.	14
1.22	GNU find, xargs, and locate	15
1.23	Fast lexical analyzer generator	15
1.24	Pattern scanning & processing.	16
1.25	GNU C/C++/Obj-C compilers	16
1.26	GNU C/C++/Obj-C compilers.	16
1.27	GNU debugger (incomplete port)	17
1.28	GNU database manager library	17
1.29	Arbitrary precision math library.	17

1.30 GNU ADA compiler.	18
1.31 GNU grep package	18
1.32 GNU groff document formatting system	18
1.33 GNU postscript interpreter	19
1.34 GNU postscript interpreter fonts	19
1.35 GNU compressing/decompressing programs	19
1.36 C code beautifier	21
1.37 GNU spelling checker	21
1.38 Shared lib providing UNIX like environ.	22
1.39 Create IXPIPE: for ixemul.library use	23
1.40 Small but very useful EMACS like editor.	23
1.41 Viewer program similar to	24
1.42 Platform independant screen maint lib.	24
1.43 GNU C++ class library	24
1.44 Runtime math library	24
1.45 A library for amiga specific gcc develop.	25
1.46 GNU macro processor	25
1.47 POSIX compatible	26
1.48 Apply diff files	26
1.49 A UNIX ksh compatible shell for AmigaDOS	26
1.50 Practical Extraction and Report Language	26
1.51 Revision Control System	27
1.52 GNU stream editor	27
1.53 GNU shell programming utilities	27
1.54 Shell archive utils and uuencode/decode.	28
1.55 GNU Tape Archiver	28
1.56 GNU termcap library.	28
1.57 GNU documentation system	29
1.58 GNU text processing utilities	29

Chapter 1

CONTENTS

1.1 GNU and Other Utilities

G N U A N D O T H E R U T I L I T I E S

This is the root directory of a set of utilities which started with Amiga ports of important GNU (GNU's Not Unix) tools, such as the GNU C compiler, and then has gradually expanded to encompass other tools.

One of the goals of this set of utilities is to have a completely self hosting environment. I.E. that everything within it be compilable by the GNU C compiler or other provided compilers. It should be possible for the recipient of these utilities to make whatever changes or bug fixes they want in any piece of code, and then rebuild and use that fixed version (and hopefully send those changes back for integration into future releases).

This file is automatically generated from the product info files in this tree.

There are three other files that are automatically generated, which are plain text versions of the same information contained in this file, as follows:

The INDEX file provides a quick overview which contains a one-line entry for each program, including the name of the program, version number, and short description.

The LOCATE file is the same as the INDEX file, except instead of having a short description, it includes the full path to the material you wish to locate.

The CONTENTS file contains detailed information about each program, including a longer description, author information, and location in the file tree.

Generally you would only need these files if you were accessing the information without the help of AmigaGuide.

This file tree is organized into several subdirectories. For an index of the files in any subdirectory, click on the appropriate name below:

src

1.2 Index of subdirectory: src

For more information about a particular program, click on the name [↔](#)
field

below:

APlusPlus		
1.01	C++ class library for AmigaDOS functions	
autoconf		
2.1	GNU automatic configuration generator.	
bc		
1.03	GNU arbitrary precision calculator lang.	
binutils		
1.8.x	GNU binary file utilities.	
binutils		
2.5.2	GNU binary utilities, Amiga source	
bison		
1.22	GNU parser generator yacc replacement	
Brik		
2.0	Compute & use CRC lists to verify files	
calc		
2.02c	Advanced desk calculator in EMACS elisp.	
cpio		
2.3	GNU utility to copy to/from archives.	
dbmalloc		
1.14	Debugging malloc/realloc/free routines.	
debug		
2.3	Macro based debugging library and tools.	
dc		
0.2	GNU reverse-polish (RPN) desk calculator	
diffutils		
2.7	GNU diff, diff3, sdiff and cmp utilities	
doschk		
1.1	Check DOS/SYSV filename limits.	
ed		
0.1	8-bit-clean POSIX compliant line editor.	

emacs	18.59	GNU Emacs editor
emacs	19.28	GNU Emacs editor
f2c	93.04.28	Fortran 77 to C translator
fileutils	3.12	File management utilities.
findutils	4.1	GNU find, xargs, and locate
flex	2.4.7	Fast lexical analyzer generator
gawk	2.15.5	Pattern scanning & processing.
gcc	2.3.3	GNU C/C++/Obj-C compilers
gcc	2.6.2	GNU C/C++/Obj-C compilers.
gdb	4.12	GNU debugger (incomplete port)
gdbm	1.7.3	GNU database manager library
ghostscript	2.6.1.4	GNU postscript interpreter
ghostscript-font	2.6.1	GNU postscript interpreter fonts
gmp	1.3.2	Arbitrary precision math library.
gnat	2.00	GNU ADA compiler.
grep	2.0	GNU grep package
groff	1.09	GNU groff document formatting system
gzip	1.2.4	GNU compressing/decompressing programs
indent	1.9.1	C code beautifier

ispell		
	4.0	GNU spelling checker
ixemul		
	40.4	Shared lib providing UNIX like environ.
ixpipe		
	1.0	Create IXPIPE: for ixemul.library use
jove		
	4.14.6	Small but very useful EMACS like editor.
less		
	252	Viewer program similar to "more".
libcurses		
	8.3	Platform independant screen maint lib.
libg++		
	2.6.1	GNU C++ class library
libm		
	5.4	Runtime math library
libnix		
	0.7	A library for amiga specific gcc develop.
m4		
	1.4	GNU macro processor
make		
	3.71	POSIX compatible "make" program
patch		
	2.1	Apply diff files
pdksh		
	4.9	A UNIX ksh compatible shell for AmigaDOS
perl		
	4.036	Practical Extraction and Report Language
rsc		
	5.6.0.1	Revision Control System
sed		
	2.05	GNU stream editor
sh-utils		
	1.12	GNU shell programming utilities
sharutils		
	4.1	Shell archive utils and uuencode/decode.
tar		
	1.11.2	GNU Tape Archiver

termcap	
1.2	GNU termcap library.
texinfo	
3.1	GNU documentation system
textutils	
1.11	GNU text processing utilities

1.3 C++ class library for AmigaDOS functions

APlusPlus 1.01 C++ class library for AmigaDOS functions

The A++ Library is a library written in C++ that consists of classes assigned to the task of encapsulating the Amiga® system software. Its aim is to provide a stable and effective method of programming the Amiga®. That should include Exec, Dos, Intuition, ARexx, ..just everything that makes sense being encapsulated with a proper C++ class interface.

The A++ Library collects the hundreds of system functions, groups them into classes that 'know' about the specialities of each function and thus shields the programmer from using them incorrectly. The C++ programming language provides the object oriented paradigm that holds no limitations to evolution of existing solutions.

The contents of this archive do not represent a complete and ready to use tool but the expression of an idea showing promise. It gives only a slight impression of what is possible. But much more can be done.

Author: Armin Vogt

Path: src/amiga/APlusPlus-1.01-src.lha

1.4 GNU automatic configuration generator.

autoconf 2.1 GNU automatic configuration generator.

Autoconf is an extensible package of m4 macros that produce shell scripts to automatically configure software source code packages. These scripts can adapt the packages to many kinds of UNIX-like systems without manual user intervention. Autoconf creates a configuration script for a package from a template file that lists the operating system features that the package can use, in the form of m4 macro calls.

Autoconf requires GNU m4. The configuration scripts produced by Autoconf are independent of Autoconf when they are run, so their users do not need to have Autoconf (or GNU m4).

Author: Free Software Foundation

Path: src/amiga/autoconf-2.1-src.lha

1.5 GNU arbitrary precision calculator lang.

bc 1.03 GNU arbitrary precision calculator lang.

A language that supports arbitrary precision numbers with interactive execution of statements. There are some similarities in the syntax to the C programming language.

A standard math library is available by command line option. If requested, the math library is defined before processing any files. bc starts by processing code from all the files listed on the command line in the order listed. After all files have been processed, bc reads from the standard input. All code is executed as it is read. (If a file contains a command to halt the processor, bc will never read from the standard input.)

This version of bc contains several extensions beyond traditional bc implementations and the POSIX draft standard. Command line options can cause these extensions to print a warning or to be rejected. This document describes the language accepted by this processor. Extensions will be identified as such.

Author: Philip A. Nelson
Path: src/amiga/bc-1.03-src.lha

1.6 GNU binary file utilities.

binutils 1.8.x GNU binary file utilities.

Various tools for operating on object and executable files. Includes "ld", "size", "nm", "strip", "ar", "objdump", and "ranlib".

Author: Free Software Foundation
Path: src/amiga/binutils-1.8.x-src.lha

1.7 GNU binary utilities, Amiga source

binutils 2.5.2 GNU binary utilities, Amiga source

The GNU binary utilities.

Author: (null)
Path: src/amiga/binutils-2.5.2-src.lha

1.8 GNU parser generator yacc replacement

bison 1.22 GNU parser generator yacc replacement

Bison is a parser generator in the style of yacc (1). It should be

upwardly compatible with input files designed for yacc. Input files should follow the yacc convention of ending in ".y". Unlike yacc, the generated files do not have fixed names, but instead use the prefix of the input file. For instance, a grammar description file named parse.y would produce the generated parser in a file named parse.tab.c, instead of yacc 's y.tab.c.

Author: Free Software Foundation
Path: src/amiga/bison-1.22-src.lha

1.9 Compute & use CRC lists to verify files

Brik 2.0 Compute & use CRC lists to verify files

A general purpose program that calculates both text and binary cyclic redundancy codes (CRCs). Text mode CRCs calculated by brik are portable across systems for files that are in the usual text format on each system. Binary mode CRCs are portable for files that are moved from system to system without any change. Brik can be used to verify and update an embedded checksum header in files. It runs under MS-DOS, UNIX system V, BSD UNIX, VAX/VMS, and AmigaDOS.

Author: Rahul Dhesi
Path: src/amiga/brik-2.0-src.lha

1.10 Advanced desk calculator in EMACS elisp.

calc 2.02c Advanced desk calculator in EMACS elisp.

"Calc" is an advanced calculator and mathematical tool that runs as part of the GNU Emacs environment. Very roughly based on the HP-28/48 series of calculators, its many features include:

- * Choice of algebraic or RPN (stack-based) entry of calculations.
 - * Arbitrary precision integers and floating-point numbers.
 - * Arithmetic on rational numbers, complex numbers (rectangular and polar), error forms with standard deviations, open and closed intervals, vectors and matrices, dates and times, infinities, sets, quantities with units, and algebraic formulas.
 - * Mathematical operations such as logarithms and trigonometric functions.
 - * Programmer's features (bitwise operations, non-decimal numbers).
 - * Financial functions such as future value and internal rate of return.
 - * Number theoretical features such as prime factorization and arithmetic modulo M for any M.
 - * Algebraic manipulation features, including symbolic calculus.
-

- * Moving data to and from regular editing buffers.
- * "Embedded mode" for manipulating Calc formulas and data directly inside any editing buffer.
- * Graphics using GNUPLOT, a versatile (and free) plotting program.
- * Easy programming using keyboard macros, algebraic formulas, algebraic rewrite rules, or extended Emacs Lisp.

Author: Dave Gillespie

Path: src/amiga/calc-2.02c-src.lha

1.11 GNU utility to copy to/from archives.

cpio 2.3 GNU utility to copy to/from archives.

Cpio copies files into or out of a cpio or tar archive, which is a file that contains other files plus information about them, such as their pathname, owner, timestamps, and access permissions. The archive can be another file on the disk, a magnetic tape, or a pipe. Cpio has three operating modes. In copy-out mode, cpio copies files into an archive. It reads a list of filenames, one per line, on the standard input, and writes the archive onto the standard output. A typical way to generate the list of filenames is with the find command; you should give find the `-depth` option to minimize problems with permissions on directories that are unwritable or not searchable.

In copy-in mode, cpio copies files out of an archive or lists the archive contents. It reads the archive from the standard input. Any non-option command line arguments are shell globbing patterns; only files in the archive whose names match one or more of those patterns are copied from the archive. Unlike in the shell, an initial `.'` in a filename does match a wildcard at the start of a pattern, and a `'/'` in a filename can match wildcards. If no patterns are given, all files are extracted.

In copy-pass mode, cpio copies files from one directory tree to another, combining the copy-out and copy-in steps without actually using an archive. It reads the list of files to copy from the standard input; the directory into which it will copy them is given as a non-option argument.

Cpio supports the following archive formats: binary, old ASCII, new ASCII, crc, HPUX binary, HPUX old ASCII, old tar, and POSIX.1 tar. The binary format is obsolete because it encodes information about the files in a way that is not portable between different machine architectures. The old ASCII format is portable between different machine architectures, but should not be used on file systems with more than 65536 i-nodes. The new ASCII format is portable between different machine architectures and can be used on any size file system, but is not supported by all versions of cpio; currently, it is only supported by GNU and Unix System V R4. The crc format is like the new ASCII format, but also contains a checksum for each file which

cpio calculates when creating an archive and verifies when the file is extracted from the archive. The HPUX formats are provided for compatibility with HPUX's cpio which stores device files differently.

The tar format is provided for compatibility with the tar program. It can not be used to archive files with names longer than 100 characters, and can not be used to archive "special" (block or character devices) files. The POSIX.1 tar format can not be used to archive files with names longer than 255 characters (less unless they have a "/" in just the right place).

By default, cpio creates binary format archives, for compatibility with older cpio programs. When extracting from archives, cpio automatically recognizes which kind of archive it is reading and can read archives created on machines with a different byte-order.

Some of the options to cpio apply only to certain operating modes; see the SYNOPSIS section for a list of which options are allowed in which modes.

Author: Phil Nelson
David MacKenzie
John Oleynick
Path: src/amiga/cpio-2.3-src.lha

1.12 Debugging malloc/realloc/free routines.

dbmalloc 1.14 Debugging malloc/realloc/free routines.

This is the malloc debugging library (patchlevel 14). This library contains a drop-in replacement for the standard dynamic memory allocation routines and many of the other C modules that are typically used to access and/or manipulate these data areas. The replacements function almost identically to the original functions with the exception that they perform a significant amount of error checking and/or validating.

This library can be used with very little (if any) changes to the original code (many times only requiring a re-link of the executable) and provide a full-service debugging capability.

The entire library will follow in 10 separate shar files.

This new version includes several enhancements over patch level 13 including:

- * more auto-configuration tuning
- * several bug fixes

In addition to the changes listed above, this version contains the following changes since the comp.sources.reviewed release at patch level 7, in Volume 2, Issue 1:

- * Much better performance
 - * automatic configuration on most systems
 - * addition of XtAlloc routines for X window debugging
 - * better identification of the source of double frees
-

- * ability to record function stack info so error reports give the calling stack for where the data was allocated and/or the error was detected (note: this is manually maintained by the pgm)
- * more support for leak detection (especially with respect to getting rid of leak reports on things you know are not leaks)
- * additional checking of pointers to make the library more robust
- * added case insensitive string comparison routines
- * several bug fixes and/or portability changes
- * added ability to change the state of the fill area flag
- * added capability to disallow reuse of free'd segments (so that double frees could be more readily identified by the data in the segment)
- * a couple of function name changes
- * new funcs: memalign, malloc_abort, malloc_size

Author: Conor P. Cahill

Path: src/amiga/dbmalloc-1.14-src.lha

1.13 Macro based debugging library and tools.

dbug 2.3 Macro based debugging library and tools.

A macro based C debugging package which has proven to be a very flexible and useful tool for debugging, testing, and porting C programs.

All of the features of the dbug package can be enabled or disabled dynamically at execution time. This means that production programs will run normally when debugging is not enabled, and eliminates the need to maintain two separate versions of a program.

Many of the things easily accomplished with conventional debugging tools, such as symbolic debuggers, are difficult or impossible with this package, and vice versa. Thus the dbug package should not be thought of as a replacement or substitute for other debugging tools, but simply as a useful addition to the program development and maintenance environment.

Author: Fred Fish

Path: src/amiga/dbug-2.3-src.lha

1.14 GNU reverse-polish (RPN) desk calculator

dc 0.2 GNU reverse-polish (RPN) desk calculator

DC is a reverse-polish desk calculator which supports unlimited precision arithmetic. It also allows you to define and call macros. Normally DC reads from the standard input; if any command arguments are given to it, they are filenames, and DC reads and executes the contents of the files before reading from standard input. All output is to standard output.

Author: Free Software Foundation
Path: src/amiga/dc-0.2-src.lha

1.15 GNU diff, diff3, sdiff and cmp utilities

diffutils 2.7 GNU diff, diff3, sdiff and cmp utilities

This directory contains the GNU diff, diff3, sdiff, and cmp utilities. Their features are a superset of the Unix features and they are significantly faster. cmp has been moved here from the GNU textutils.

Computer users often find occasion to ask how two files differ. Perhaps one file is a newer version of the other file. Or maybe the two files started out as identical copies but were changed by different people.

You can use the diff command to show differences between two files, or each corresponding file in two directories. diff outputs differences between files line by line in any of several formats, selectable by command line options. This set of differences is often called a diff or patch. For files that are identical, diff normally produces no output; for binary (non-text) files, diff normally reports only that they are different.

You can use the cmp command to show the offsets and line numbers where two files differ. Cmp can also show all the characters that differ between the two files, side by side.

You can use the diff3 command to show differences among three files. When two people have made independent changes to a common original, diff3 can report the differences between the original and the two changed versions, and can produce a merged file that contains both persons' changes together with warnings about conflicts.

You can use the sdiff command to merge two files interactively.

You can use the set of differences produced by diff to distribute updates to text files (such as program source code) to other people. This method is especially useful when the differences are small compared to the complete files. Given diff output, you can use the patch program to update, or patch, a copy of the file. If you think of diff as subtracting one file from another to produce their difference, you can think of patch as adding the difference to one file to reproduce the other.

Author: Free Software Foundation
et. al.
Path: src/amiga/diffutils-2.7-src.lha

1.16 Check DOS/SYSV filename limits.

doschk 1.1 Check DOS/SYSV filename limits.

This program is intended as a utility to help software developers ensure that their source file names are distinguishable on MS-DOS and 14-character SYSV platforms. To perform this task, doschk reads a list of filenames and produces a report of all the conflicts that would arise if the files were transferred to a MS-DOS or SYSV platform.

Author: DJ Delorie

Path: src/amiga/doschk-1.1-src.lha

1.17 8-bit-clean POSIX compliant line editor.

ed 0.1 8-bit-clean POSIX compliant line editor.

"Ed" is a line-oriented text editor. It is used to create, display, modify and otherwise manipulate text files. "Red" is a restricted ed: it can only edit files in the current directory and cannot execute shell commands.

Author: Andrew Moore

Path: src/amiga/ed-0.1-src.lha

1.18 GNU Emacs editor

emacs 18.59 GNU Emacs editor

GNU Emacs is the GNU incarnation of the advanced, self-documenting, customizable, extensible real-time display editor Emacs. (The 'G' in 'GNU' is not silent.)

We say that Emacs is a "display" editor because normally the text being edited is visible on the screen and is updated automatically as you type your commands.

We call it a "real-time" editor because the display is updated very frequently, usually after each character or pair of characters you type. This minimizes the amount of information you must keep in your head as you edit.

We call Emacs advanced because it provides facilities that go beyond simple insertion and deletion: filling of text; automatic indentation of programs; viewing two or more files at once; and dealing in terms of characters, words, lines, sentences, paragraphs, and pages, as well as expressions and comments in several different programming languages. It is much easier to type one command meaning "go to the end of the paragraph" than to find that spot with simple cursor keys.

"Self-documenting" means that at any time you can type a special character, 'Control-h', to find out what your options are. You can

also use it to find out what any command does, or to find all the commands that pertain to a topic.

"Customizable" means that you can change the definitions of Emacs commands in little ways. For example, if you use a programming language in which comments start with '`**`' and end with '`**>`', you can tell the Emacs comment manipulation commands to use those strings. Another sort of customization is rearrangement of the command set. For example, if you prefer the four basic cursor motion commands (up, down, left and right) on keys in a diamond pattern on the keyboard, you can have it.

"Extensible" means that you can go beyond simple customization and write entirely new commands, programs in the Lisp language to be run by Emacs's own Lisp interpreter. Emacs is an "on-line extensible" system, which means that it is divided into many functions that call each other, any of which can be redefined in the middle of an editing session. Any part of Emacs can be replaced without making a separate copy of all of Emacs. Most of the editing commands of Emacs are written in Lisp already; the few exceptions could have been written in Lisp but are written in C for efficiency. Although only a programmer can write an extension, anybody can use it afterward.

Author: Richard Stallman
Path: src/amiga/emacs-18.59-src.lha

1.19 GNU Emacs editor

emacs 19.28 GNU Emacs editor

GNU Emacs is the GNU incarnation of the advanced, self-documenting, customizable, extensible real-time display editor Emacs. (The '`G`' in '`GNU`' is not silent.)

We say that Emacs is a "display" editor because normally the text being edited is visible on the screen and is updated automatically as you type your commands.

We call it a "real-time" editor because the display is updated very frequently, usually after each character or pair of characters you type. This minimizes the amount of information you must keep in your head as you edit.

We call Emacs advanced because it provides facilities that go beyond simple insertion and deletion: filling of text; automatic indentation of programs; viewing two or more files at once; and dealing in terms of characters, words, lines, sentences, paragraphs, and pages, as well as expressions and comments in several different programming languages. It is much easier to type one command meaning "go to the end of the paragraph" than to find that spot with simple cursor keys.

"Self-documenting" means that at any time you can type a special character, '`Control-h`', to find out what your options are. You can also use it to find out what any command does, or to find all the commands that pertain to a topic.

"Customizable" means that you can change the definitions of Emacs commands in little ways. For example, if you use a programming language in which comments start with `<*>` and end with `*>`, you can tell the Emacs comment manipulation commands to use those strings. Another sort of customization is rearrangement of the command set. For example, if you prefer the four basic cursor motion commands (up, down, left and right) on keys in a diamond pattern on the keyboard, you can have it.

"Extensible" means that you can go beyond simple customization and write entirely new commands, programs in the Lisp language to be run by Emacs's own Lisp interpreter. Emacs is an "on-line extensible" system, which means that it is divided into many functions that call each other, any of which can be redefined in the middle of an editing session. Any part of Emacs can be replaced without making a separate copy of all of Emacs. Most of the editing commands of Emacs are written in Lisp already; the few exceptions could have been written in Lisp but are written in C for efficiency. Although only a programmer can write an extension, anybody can use it afterward.

Author: Richard Stallman
Path: src/amiga/emacs-19.28-src.lha

1.20 Fortran 77 to C translator

f2c 93.04.28 Fortran 77 to C translator

F2c converts Fortran 77 source code in files with names ending in `.f` or `.F` to C (or C++) source files in the current directory, with `.c` substituted for the final `.f` or `.F`. If no Fortran files are named, f2c reads Fortran from standard input and writes C on standard output. Filenames that end with `.p` or `.P` are taken to be prototype files, as produced by option `-P`, and are read first.

Author: David Gay
Stu Feldman
Mark Maimone
Norm Schryer
Path: src/amiga/f2c-1993.04.28-src.lha

1.21 File management utilities.

fileutils 3.12 File management utilities.

These are the GNU file management utilities. Most of these programs have significant advantages over their Unix counterparts, such as greater speed, additional options, and fewer arbitrary limits.

The programs that can be built with this package are: `chgrp`, `chown`, `chmod`, `cp`, `dd`, `df`, `du`, `install`, `ln`, `dir`, `vdir`, `ls`, `mkdir`, `mkdir`, `mkfifo`, `mknod`, `mv`, `rm`, `rmdir` and `touch`. But `mkdir` is built only on

systems that lack the rename system call.

The programs `cat`, `cut`, `expand`, `head`, `paste`, `split`, `tac`, `tail` and `unexpand`, which used to be part of the `fileutils`, are now part of the `textutils`. `Cmp` is now part of the `diff` distribution.

Author: Free Software Foundation
Path: `src/amiga/fileutils-3.12-src.lha`

1.22 GNU find, xargs, and locate

`findutils` 4.1 GNU find, xargs, and locate

This package contains the GNU `find`, `xargs`, and `locate` programs. This version of `find` and `xargs` comply with POSIX 1003.2, and also support some additional options, some borrowed from Unix and some unique to GNU.

To gain speed, GNU `find` now avoids statting files whenever possible. It does this by: (1) Checking the number of links to directories and not statting files that it knows aren't directories until it encounters a test or action that needs the stat info. (2) Rearranging the command line, where possible, so that it can do tests that don't require a stat before tests that do, in hopes that the latter will be skipped because of an OR or AND. (But it only does this where it will leave the output unchanged.)

The `locate` utility is based on James Woods' public domain fast-find code, which is also distributed with the 4.3BSD `find`. Because POSIX requires `'find foo'` to have the same effect as `'find foo -print'`, the fast-find searching has been moved to a separate program, `'locate'`; the same thing has been done in 4.3BSD-reno/4.4BSD. If you use `locate`, you should run the included `'updatedb'` script from cron periodically (typically nightly).

Author: Free Software Foundation
Path: `src/amiga/findutils-4.1-src.lha`

1.23 Fast lexical analyzer generator

`flex` 2.4.7 Fast lexical analyzer generator

Flex is a tool for generating scanners, programs which recognized lexical patterns in text. Flex reads the given input files, or its standard input if no file names are given, for a description of a scanner to generate. The description is in the form of pairs of regular expressions and C code, called rules. Flex generates as output a C source file, `lex.yy.c`, which defines a routine `yylex()`. This file is compiled and linked with the `-lfl` library to produce an executable. When the executable is run, it analyzes its input for occurrences of the regular expressions. Whenever it finds one, it executes the corresponding C code.

Author: Vern Paxson
Van Jacobson
Jef Poskanzer
Path: src/amiga/flex-2.4.7-src.lha

1.24 Pattern scanning & processing.

gawk 2.15.5 Pattern scanning & processing.

Gawk is the GNU Project's implementation of the AWK programming language. It conforms to the definition of the language in the POSIX 1003.2 Command Language And Utilities Standard. This version in turn is based on the description in "The AWK Programming Language", by Aho, Kernighan, and Weinberger, with the additional features defined in the System V Release 4 version of UNIX awk. Gawk also provides some GNU-specific extensions.

The command line consists of options to gawk itself, the AWK program text (if not supplied via the `-f` or `--file` options), and values to be made available in the ARGV and ARGV pre-defined AWK variables.

Author: Paul Rubin
Jay Fenlason
Path: src/amiga/gawk-2.15.5-src.lha

1.25 GNU C/C++/Obj-C compilers

gcc 2.3.3 GNU C/C++/Obj-C compilers

The GNU C, C++, and Objective C compilers. Includes all support for compiling C, C++ and Objective C, including a run-time library for Objective C.

Author: Free Software Foundation
Path: src/amiga/gcc-2.3.3-src.lha

1.26 GNU C/C++/Obj-C compilers.

gcc 2.6.2 GNU C/C++/Obj-C compilers.

The GNU C, C++, and Objective C compilers. Includes all support for compiling C, C++ and Objective C, including a run-time library for Objective C.

Author: Free Software Foundation
Path: src/amiga/gcc-2.6.3-src.lha

1.27 GNU debugger (incomplete port)

`gdb` 4.12 GNU debugger (incomplete port)

The purpose of a debugger such as GDB is to allow you to see what is going on "inside" another program while it executes, or what another program was doing at the moment it crashed.

GDB can do four main kinds of things (plus other things in support of these) to help you catch bugs in the act: (1) start your program, specifying anything that might affect its behavior, (2) make your program stop on specified conditions, (3) examine what has happened, when your program has stopped, (4) change things in your program, so you can experiment with correcting the effects of one bug and go on to learn about another.

You can use GDB to debug programs written in C, C++, and Modula-2. Fortran support will be added when a GNU Fortran compiler is ready.

Author: Free Software Foundation
Path: `src/amiga/gdb-4.12-src.lha`

1.28 GNU database manager library

`gdbm` 1.7.3 GNU database manager library

GNU dbm is a library of routines that manages data files that contain key/data pairs. The access provided is that of storing, retrieval, and deletion by key and a non-sorted traversal of all keys. A process is allowed to use multiple data files at the same time.

Author: Philip A. Nelson
Path: `src/amiga/gdbm-1.7.3-src.lha`

1.29 Arbitrary precision math library.

`gmp` 1.3.2 Arbitrary precision math library.

GNU MP is a library for arbitrary precision arithmetic, operating on signed integers and rational numbers. It has a rich set of functions, and the functions have a regular interface.

The author has tried to make these functions as fast as possible, both for small operands and for huge operands. The speed is achieved by using fullwords as the basic arithmetic type, by using fast algorithms, by defining inline assembler for mixed sized multiplication and division (i.e 32*32->64 bit multiplication and 64/32->32,32 bit division), and by hacking the code with emphasis on speed (and not simplicity and elegance).

The speed of GNU MP is about 5 to 100 times that of Berkeley MP for small operands. The speed-up increases with the operand sizes for

certain operations, for which GNU MP has asymptotically faster algorithms.

Author: Torbjorn Granlund
Path: src/amiga/gmp-1.3.2-src.lha

1.30 GNU ADA compiler.

gnat 2.00 GNU ADA compiler.

The GNU ADA compiler. Includes all support for compiling ADA.

Author: Free Software Foundation
Path: src/amiga/gnat-2.00-src.lha

1.31 GNU grep package

grep 2.0 GNU grep package

GNU grep is based on a fast lazy-state deterministic matcher (about twice as fast as stock Unix egrep) hybridized with a Boyer-Moore-Gosper search for a fixed string that eliminates impossible text from being considered by the full regexp matcher without necessarily having to look at every character. The result is typically many times faster than Unix grep or egrep. (Regular expressions containing backreferencing will run more slowly, however.)

Author: Mike Haertel
Arthur David Olson
Richard Stallman
Karl Berry
Henry Spencer
Scott Anderson
David MacKenzie
James Woods
Andrew Hume
Path: src/amiga/grep-2.0-src.lha

1.32 GNU groff document formatting system

groff 1.09 GNU groff document formatting system

This is the GNU groff document formatting system. Included in this release are implementations of troff, pic, eqn, tbl, refer, the -man macros and the -ms macros, and drivers for PostScript, TeX dvi format, and typewriter-like devices. Also included is a modified version of the Berkeley -me macros, an enhanced version of the X11 xditview previewer, and an implementation of the -mm macros.

Author: James Clark

Path: src/amiga/groff-1.09-src.lha

1.33 GNU postscript interpreter

ghostscript 2.6.1.4 GNU postscript interpreter

Ghostscript is the name of a set of software that provides an interpreter for the PostScript (TM) language, and a set of C procedures (the Ghostscript library) that implement the graphics capabilities that appear as primitive operations in the PostScript language.

Author: L. Peter Deutsch

Path: src/amiga/gs-2.6.1.4-src.lha

1.34 GNU postscript interpreter fonts

ghostscript-fonts 2.6.1 GNU postscript interpreter fonts

Ghostscript is the name of a set of software that provides an interpreter for the PostScript (TM) language, and a set of C procedures (the Ghostscript library) that implement the graphics capabilities that appear as primitive operations in the PostScript language.

Author: L. Peter Deutsch

Path: src/amiga/gs-fonts-2.6.1-src.lha

1.35 GNU compressing/decompressing programs

gzip 1.2.4 GNU compressing/decompressing programs

Gzip reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension .gz, while keeping the same ownership modes, access and modification times. (The default extension is -gz for VMS, z for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. Gzip will only attempt to compress regular files. In particular, it will ignore symbolic links.

If the compressed file name is too long for its file system, gzip truncates it. Gzip attempts to truncate only the parts of the file name longer than 3 characters. (A part is delimited by dots.) If the name consists of small parts only, the longest parts are truncated. For example, if file names are limited to 14 characters, gzip.msdos.exe is compressed to gzi.msdx.exe.gz. Names are not truncated on systems which do not have a limit on file name length.

By default, gzip keeps the original file name and timestamp in the

compressed file. These are used when decompressing the file with the `-N` option. This is useful when the compressed file name was truncated or when the time stamp was not preserved after a file transfer.

Compressed files can be restored to their original form using `gzip -d` or `gunzip` or `zcat`. If the original name saved in the compressed file is not suitable for its file system, a new name is constructed from the original one to make it legal.

`gunzip` takes a list of files on its command line and replaces each file whose name ends with `.gz`, `-gz`, `.z`, `-z`, `_z` or `.Z` and which begins with the correct magic number with an uncompressed file without the original extension. `gunzip` also recognizes the special extensions `".tgz"` and `".taz"` as shorthands for `.tar.gz` and `.tar.Z` respectively. When compressing, `gzip` uses the `.tgz` extension if necessary instead of truncating a file with a `.tar` extension.

`gunzip` can currently decompress files created by `gzip`, `zip`, `compress`, `compress -H` or `pack`. The detection of the input format is automatic. When using the first two formats, `gunzip` checks a 32 bit CRC. For `pack`, `gunzip` checks the uncompressed length. The standard `compress` format was not designed to allow consistency checks. However `gunzip` is sometimes able to detect a bad `.Z` file. If you get an error when uncompressing a `.Z` file, do not assume that the `.Z` file is correct simply because the standard `uncompress` does not complain. This generally means that the standard `uncompress` does not check its input, and happily generates garbage output. The SCO `compress -H` format (`lzh` compression method) does not include a CRC but also allows some consistency checks.

Files created by `zip` can be uncompressed by `gzip` only if they have a single member compressed with the 'deflation' method. This feature is only intended to help conversion of `tar.zip` files to the `tar.gz` format. To extract `zip` files with several members, use `unzip` instead of `gunzip`.

`zcat` is identical to `gunzip -c`. (On some systems, `zcat` may be installed as `gzcat` to preserve the original link to `compress`.) `zcat` uncompresses either a list of files on the command line or its standard input and writes the uncompressed data on standard output. `zcat` will uncompress files that have the correct magic number whether they have a `.gz` suffix or not.

`Gzip` uses the Lempel-Ziv algorithm used in `zip` and `PKZIP`. The amount of compression obtained depends on the size of the input and the distribution of common substrings. Typically, text such as source code or English is reduced by 60-70%. Compression is generally much better than that achieved by `LZW` (as used in `compress`), Huffman coding (as used in `pack`), or adaptive Huffman coding (`compact`).

Compression is always performed, even if the compressed file is slightly larger than the original. The worst case expansion is a few bytes for the `gzip` file header, plus 5 bytes every 32K block, or an expansion ratio of 0.015% for large files. Note that the actual number of used disk blocks almost never increases. `gzip` preserves the mode, ownership and timestamps of files when compressing or decompressing.

Author: Jean-loup Gailly
Path: src/amiga/gzip-1.2.4-src.lha

1.36 C code beautifier

indent 1.9.1 C code beautifier

The 'indent' program can be used to make code easier to read. It can also convert from one style of writing C to another, and understands a substantial amount about the syntax of C, but it also attempts to cope with incomplete and misformed syntax. In version 1.2 and more recent versions, the GNU style of indenting is the default.

Author: Free Software Foundation
Path: src/amiga/indent-1.9.1-src.lha

1.37 GNU spelling checker

ispell 4.0 GNU spelling checker

Ispell is a program that helps you to correct typos in a file, and to find the correct spelling of words. When presented with a word that is not in the dictionary, ispell attempts to find near misses that might include the word you meant.

The best way to use ispell is with GNU EMACS. For documentation about this mode, see the info topic "ispell".

Ispell can also be used by itself, and in this case the most common usage is "ispell filename". If ispell finds a word that is not in the dictionary, it is printed at the top of the screen. Ispell then checks the dictionary for near misses \- words that differ only by a single letter, a missing or extra letter, or a pair of transposed letters. Any that are found are printed on the following lines, and finally, two lines of context containing the word are printed at the bottom of the screen. If your terminal can type in reverse video, the word itself is highlighted.

If you think the word is correct as it stands, you can type either "Space" to accept it this one time, "A" to accept it for the rest of this file, or "I" to accept it and put it in your private dictionary. If one of the near misses is the word you want, type the corresponding number. Finally, you can type "R" and you will be prompted for a replacement word. The string you type will be broken into words, and each one will also be checked. You can also type "?" for help.

If ispell is started with no arguments, it enters a loop reading words from the standard input, and printing messages about them on the standard output. You can use this mode to find the spelling of a problem word.

Author: Pace Willisson

Path: src/amiga/ispell-4.0-src.lha

1.38 Shared lib providing UNIX like environ.

ixemul 40.4 Shared lib providing UNIX like environ.

(by Markus Wild)

I always wanted a library, that would emulate as much as possible of a **IX/BSD environment on the Amiga, so that programs (usually programming tools) written for **IX/BSD could be ported in a quick and straight forward way to the Amiga. I guess the library accomplishes this goal fairly well.

What it is and what it isn't

=====

The design of the library was therefore guided towards **IX/BSD compatibility, and *not* :

- o to be too conservative with resources
- o to be particularly conformant to Amiga habits. Thus if I had to decide whether I should make a function act more like an Amiga function or more like a **IX/BSD one, I decided for the latter. As an example: `_cli_parse()` does wildcard expansion, and tries to apply more or less **IX/BSD shell semantics to an argument line, it doesn't call `ReadArgs()`.

The types used in my own source code are all from `sys/types.h` (except `BPTR`). I don't think capitalized identifiers should be used for typedef'd types. According to C-conventions, anything written in captials should be `#undef'inable`, which typedefs aren't. Thus if you write contributions to be included into the official distribution of this library, code according to this. Use `'u_char'` and not `UBYTE`, etc. I don't care that this is against the Commodore coding standard, this is my code, and I decide what I like and what not.

- o to be particularly suited for inclusion into a shared library, although most things *are* shared now. What I'd really want for the Amiga is the concept of a dynamic linker.

On the other hand, it should be:

- o expandable. As an example, a file descriptor already can refer to 'real' files, directories, memory buffers treated as files. I plan to add sockets in some next release (Commodore: please get out some examples on how to use SANA-II stuff, so my sockets can be compatible!)
- o patchable. If you want some function to behave differently, you can `SetFunction()` it, and the rest of the library should use your new entry. NOTE: I used this only for major functions, that may reasonably change. I didn't call functions like `strcmp()`, `strlen()`, `bcopy()` that way for efficiency reasons (and my lazyness to change the whole string/ and other libraries;-))

This version doesn't particularly follow this goal very well, mostly `stdio` is still the original BSD code, and doesn't use `syscall()`...

Author: Markus M. Wild

Path: src/amiga/ixemul-40.4-src.lha

1.39 Create IXPIPE: for ixemul.library use

ixpipe 1.0 Create IXPIPE: for ixemul.library use

A pipe handler for use with ixemul.library.

Author: Markus Wild

Path: src/amiga/ixpipe-1.0-src.lha

1.40 Small but very useful EMACS like editor.

jove 4.14.6 Small but very useful EMACS like editor.

JOVE stands for Jonathan's Own Version of Emacs.

JOVE is an advanced, self-documenting, customizable real-time display editor. It (and this tutorial introduction) are based on the original EMACS editor and user manual written at M.I.T. by Richard Stallman+.

Although JOVE is meant to be compatible with EMACS, and indeed many of the basic commands are very similar, there are some major differences between the two editors, and you should not rely on their behaving identically.

JOVE is considered a display editor because normally the text being edited is visible on the screen and is updated automatically as you type your commands.

It's considered a real-time editor because the display is updated very frequently, usually after each character or pair of characters you type. This minimizes the amount of information you must keep in your head as you edit.

JOVE is advanced because it provides facilities that go beyond simple insertion and deletion: filling of text; automatic indentations of programs; view more than one file at once; and dealing in terms of characters, words, lines, sentences and paragraphs. It is much easier to type one command meaning "go to the end of the paragraph" than to find the desired spot with repetition of simpler commands.

Self-documenting means that at almost any time you can easily find out what a command does, or to find all the commands that pertain to a topic.

Customizable means that you can change the definition of JOVE commands in little ways. For example, you can rearrange the command set; if you prefer to use arrow keys for the four basic cursor motion commands (up, down, left and right), you can. Another sort of customization is writing new commands by combining built in commands.

Author: Jonathan Payne

Path: src/amiga/jove-4.14.6-src.lha

1.41 Viewer program similar to

less 252 Viewer program similar to "more".

Less is a program similar to more (1), but which allows backward movement in the file as well as forward movement. Also, less does not have to read the entire input file before starting, so with large input files it starts up faster than text editors like vi (1). Less uses termcap (or terminfo on some systems), so it can run on a variety of terminals. There is even limited support for hardcopy terminals. (On a hardcopy terminal, lines which should be printed at the top of the screen are prefixed with a caret.)

Commands are based on both more and vi. Commands may be preceded by a decimal number, called N in the descriptions below. The number is used by some commands, as indicated.

Author: Mark Nudelman et al.
Path: src/amiga/less-252-src.lha

1.42 Platform independant screen maint lib.

libcurses 8.3 Platform independant screen maint lib.

In making available the generalized terminal descriptions in termcap, much information was made available to the programmer, but little work was taken out of one's hands. The purpose of this package is to allow the C programmer to do the most common type of terminal dependent functions, those of movement optimization and optimal screen updating, without doing any of the dirty work, and with nearly as much ease as is necessary to simply print or read things.

Author: Kenneth C. R. C. Arnold
Elan Amir
Path: src/amiga/libcurses-8.3-src.lha

1.43 GNU C++ class library

libg++ 2.6.1 GNU C++ class library

A C++ class library for use with the GNU C++ compiler.

Author: (null)
Path: src/amiga/libg++-2.6.2-src.lha

1.44 Runtime math library

libm 5.4 Runtime math library

This runtime math library is from BSD. It is used with the GNU C compiler when the `-lm` option is given to gcc.

Author: (null)

Path: `src/amiga/libm-5.4-src.lha`

1.45 A library for amiga specific gcc develop.

`libnix` 0.7 A library for amiga specific gcc develop.

This is `libnix`, a static (i.e. link) library for gcc 2.3.3 or above. It's not a replacement for `ixemul.library` (though it's possible to recompile most of the gcc environment with `libnix`) but a good thing for amiga specific development on gcc:

- * It's mostly compatible to SAS's way of handling things, i.e. you get even an automatic shared library opening feature and some other things you may miss in `ixemul.library`. This also means it's ANSI compliant.
- * It doesn't need any shared libraries than normal Amiga OS ones.
- * It is not copyrighted by the FSF. Therefore you neither need to include sources nor objects together with your executable. (read the GLGPL `_before_` flaming on this statement)
- * And it's short! I was able to compile a 492 byte 'hello, world' using normal main.
- * It uses OS20 features whenever necessary.

To cut it short:

Use `ixemul.library` for porting Un*x programs, `libnix` for compiling amiga-only programs and gcc becomes one of the best Amiga compilers.

Author: Matthias Fleischer

Gunther Nikl

Path: `src/amiga/libnix-0.7-src.lha`

1.46 GNU macro processor

`m4` 1.4 GNU macro processor

This is GNU `m4`, a program which copies its input to the output, expanding macros as it goes. `m4` has built-in functions for including named files, running Unix commands, doing integer arithmetic, manipulating text in various ways, recursion, etc... Macros can also be user-defined, and can take any number of arguments.

Author: Rene' Seindal

Path: `src/amiga/m4-1.4-src.lha`

1.47 POSIX compatible

make 3.71 POSIX compatible "make" program

The "make" utility automatically determines which pieces of a large program need to be recompiled, and issues commands to recompile them. GNU "make" conforms to section 6.2 of "IEEE Standard 1003.2-1992" (POSIX.2).

Author: Richard Stallman
Roland McGrath
Path: src/amiga/make-3.71-src.lha

1.48 Apply diff files

patch 2.1 Apply diff files

Patch will take a patch file containing any of the four forms of difference listing produced by the diff program and apply those differences to an original file, producing a patched version. By default, the patched version is put in place of the original, with the original file backed up to another name.

Author: Larry Wall
Path: src/amiga/patch-2.1-src.lha

1.49 A UNIX ksh compatible shell for AmigaDOS

pdksh 4.9 A UNIX ksh compatible shell for AmigaDOS

A KSH-like shell that is compatible enough with the real UNIX ksh to be used with most scripts that the UNIX ksh can run.

Author: Eric Gisin
Charles Forsyth
John R MacMillan
Simon J. Gerraty
Markus Wild
Path: src/amiga/pdksh-4.9-src.lha

1.50 Practical Extraction and Report Language

perl 4.036 Practical Extraction and Report Language

Perl is an interpreted language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It's also a good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny,

elegant, minimal). It combines (in the author's opinion, anyway) some of the best features of C, sed, awk, and sh, so people familiar with those languages should have little difficulty with it. (Language historians will also note some vestiges of csh, Pascal, and even BASIC-PLUS.) Expression syntax corresponds quite closely to C expression syntax.

Unlike most Unix utilities, perl does not arbitrarily limit the size of your data. If you've got the memory, perl can slurp in your whole file as a single string. Recursion is of unlimited depth. And the hash tables used by associative arrays grow as necessary to prevent degraded performance. Perl uses sophisticated pattern matching techniques to scan large amounts of data very quickly. Although optimized for scanning text, perl can also deal with binary data, and can make dbm files look like associative arrays (where dbm is available). Setuid perl scripts are safer than C programs through a dataflow tracing mechanism which prevents many stupid security holes. If you have a problem that would ordinarily use sed or awk or sh, but it exceeds their capabilities or must run a little faster, and you don't want to write the silly thing in C, then perl may be for you. There are also translators to turn your sed and awk scripts into perl scripts.

Author: Larry Wall
Path: src/amiga/perl-4.036-src.lha

1.51 Revision Control System

rsc 5.6.0.1 Revision Control System

RCS, the Revision Control System, manages multiple revisions of files. RCS can store, retrieve, log, identify, and merge revisions. It is useful for files that are revised frequently, e.g. programs, documentation, graphics, and papers.

Author: Walter F. Tichy
et. al.
Path: src/amiga/rsc-5.6.0.1-src.lha

1.52 GNU stream editor

sed 2.05 GNU stream editor

Sed copies named files, or the standard input, to the standard output, edited according to a script of commands.

Author: Free Software Foundation
Path: src/amiga/sed-2.05-src.lha

1.53 GNU shell programming utilities

sh-utils 1.12 GNU shell programming utilities

This is a package of small shell programming utilities. They are mostly compliant with POSIX.2, where applicable. The programs that can be built with this package are: basename date dirname echo env expr false groups id logname nice nohup pathchk printenv printf sleep stty tee test true tty uname who whoami yes Some programs (uname, nice, nohup, and stty) are built and installed only on systems that have the features to support them.

Author: Free Software Foundation
Path: src/amiga/sh-utils-1.12-src.lha

1.54 Shell archive utils and uuencode/decode.

sharutils 4.1 Shell archive utils and uuencode/decode.

'shar' makes so-called shell archives out of many files, preparing them for transmission by electronic mail services. 'unshar' helps unpacking shell archives after reception. The core of both programs is initially derived from public domain. Some modules and other code sections are freely borrowed from other GNU distributions, bringing 'shar' under the terms of the GNU General Public License.

'uuencode' prepares a file for transmission over an electronic channel which ignores or otherwise mangles the eight bit (high order bit) of bytes. 'uudecode' does the converse transformation. They are derived from the BSD NET/2 distribution. If you have more powerful 'uuencode' and 'uudecode' already available, you may want to use './configure --disable-uencode' to prevent their installation.

Author: Free Software Foundation
Path: src/amiga/sharutils-4.1-src.lha

1.55 GNU Tape Archiver

tar 1.11.2 GNU Tape Archiver

Tar collects files into an archive which is normally written to tape or other backup media. It can also be written to a normal file, and such files have become a common intersystem exchange mechanism.

Author: John Gilmore
Path: src/amiga/tar-1.11.2-src.lha

1.56 GNU termcap library.

termcap 1.2 GNU termcap library.

This is the GNU termcap library, a library of C functions that enable programs to send control strings to terminals in a way independent of the terminal type. Most of this package is also distributed with GNU Emacs, but it is available in this separate distribution to make it easier to install as `-ltermcap`.

Author: Free Software Foundation
Path: `src/amiga/termcap-1.2-src.lha`

1.57 GNU documentation system

`texinfo` 3.1 GNU documentation system

Texinfo is a documentation system that uses a single source file to produce both on-line information and printed output. This means that instead of writing two different documents, one for the on-line help or other on-line information and the other for a typeset manual or other printed work, you need write only one document. When the work is revised, you need revise only one document. You can read the on-line information, known as an "Info file", with an Info documentation-reading program.

Author: Free Software Foundation
Path: `src/amiga/texinfo-3.1-src.lha`

1.58 GNU text processing utilities

`textutils` 1.11 GNU text processing utilities

These are the GNU text file (actually, file contents) processing utilities. Most of these programs have significant advantages over their Unix counterparts, such as greater speed, additional options, and fewer arbitrary limits. The programs that can be built with this package are: `cat`, `cksum`, `comm`, `csplit`, `cut`, `expand`, `fold`, `head`, `join`, `nl`, `od`, `paste`, `pr`, `sort`, `split`, `sum`, `tac`, `tail`, `tr`, `unexpand`, `uniq`, and `wc`. The `cmp` program has moved to the GNU diff distribution.

Author: Free Software Foundation
Path: `src/amiga/textutils-1.11-src.lha`
