amigaMP3

COLLABORATORS			
	TITLE : amigaMP3		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		April 16, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

amigaMP3

1.1 amigaMP3.guide

```
                mpeg3play0.9.5 a1

Dodgy docs by
                 Michael Cheng

                 Introduction

                 History

                 Requirements

                 Usage

                 Libnix Version
                 The noixemul versions.

                 Speed

                 Source

                 Masochists
                 For those that want to encode mp3's.

                 Problems

                 Feedback

                 Future

                 Acknowledgements

                 Author
```

1.2 Introduction

Um. It's an mp3 decoder. It won't actually be a *player* until we all get
100Mhz 060's.

This is another in a long line of Cstarware (see the faq on Darxide).

1.3 History

```
                11 December 1996
    - first release
    - using C source from mpeg1_iis v4.1 from
                Fraunhofer Institute
                12 February 1997
    - v2.0 beta release.
    - Now using source code from
                Johan Hagman
                    - supports more mp3's
    - runs more quickly.

27 February 1997
    - v0.9.5 (now copying original source version numbering)
    - Nearly a straight compile of the unix source [which includes
  modifications for the amiga made by Troels]
    - approximately 30% faster than the previous release
    - Changed musicout.c such that you can now have AIFF sound files
  or raw PCM output. [-r switch]
    - added the compile option -mstackextend.  There are now no problems
  with low stack shells. Negligible speed penalty <1%

2 March 1997
    - v0.9.5-a1  Same source as v0.9.5, just 'a1' to indicate a new amiga version.
    - now comes in a version with doesn't use ixemul.  This libnix version
  lacks a couple of features:
      + fork()/vfork() not supported.  [probably doesn't work
    properly in the ixemul version either]. In the libnix
    version forking off a new player is commented out totally.
      + reading mp3 from stdin is not supported.
    [I'm not sure on how to fake an fdopen() which works with
    ixemul, but fails with libnix]
      + only 68881 libnix versions available
    - I RTFM for gcc and found that the 020-040 version actually requires
  an FPU. D'oh.
    - Added a new 020-nofpu version. Suitable for 030's without fpu's as well.
    - Added an mp3 encoder for the masochists out there.
```

1.4 Requirements

```
    - an amiga
    - for the ixemul versions: ixemul.library v45.1+
    - an FPU is highly recommended.
```

1.5 Usage

```
5 versions supplied. Compiled with different gcc options. Rename to suit
your needs.

    mpeg3play.000    - any amiga
    mpeg3play.020    - 020+
    mpeg3play.020881 - 020+ with FPU (-m68020 -m68881)
    mpeg3play.020040 - any processor 020+881+ (-m68020-40)
    mpeg3play.040881 - 040 with FPU

Usage:
    mpeg3play -o <aiff_output_file> <mpegaudio_file>

Or just:
    mpeg3play
  to get a list of options.

Or, if you want raw PCM output
    mpeg3play -r -o <raw_PCM_output_file> <mpegaudio_file>

Or, if you want to convert straight from mp3 to mp2

    stack 100000    (for musicin)
    run mpeg3play -r -o pipe:mp3 <mp3file>
    musicin pipe:mp3 <mp2file> -b128



Warning! Mpeg audio files are often compressed at ratios greater than 10:1,
so you better make sure you have lots of HD space to decode them.

Note: I use Play16 to play back the audio files.

    play16 file.aiff

If you use the raw (-r) option, you will need to specify the type of file when
playing back with play16. eg

    play16 freq=44100 tracks=2 bits=16 spot1.raw
```

1.6 Libnix

```
This version was inspired by Drizzit on #amiga.

Not all people like the ixemul.library.  However, it is very convenient for
porting unix software to the amiga as usually there are no changes required
to the original source.

ixemul is a relatively large library and people on low mem systems could
probably do without having to load it.  Others have an objection on principal,
similar to some people's objection to MUI.
```

Anyway, I have played with the source a bit more and produced a libnix version.
I make no guarantees for the libnix version. [I don't make any for the ixemul
version either, but I make fewer guarantees for the libnix one :)]

In particular I had to make certain changes of which I am not 100% sure. I
realise there must be better workarounds for them, but I don't have enough
experience to know what they are:

- the command fdopen() isn't part of libnix, so the ability to read
the mp3 file from stdin had to be removed
- fork() isn't a part of libnix. I'm not sure if you could fork the
ixemul version, but you definitely can't fork the libnix version
as I commented that code out completely.
- libnix refuses to open the output file in the mode "w+b", so I open
it in mode "wb". Since you can't fseek() in "wb" mode, the code
to leave space for the AIFF header no longer works. I fudged this
by making the program write the exact number of bytes that would
have been the AIFF header before starting the decoding. This
leaves the file pointer in the correct position, and is exactly
the same as if I had fseek'ed to that position. [It's a damn
ugly way of doing it though]
- libnix doesn't compile the source properly if there isn't an FPU
option. The libnix normal math library doesn't support frexp()
and ldexp(), while the libnix 68881 library does. I tried
using a different math library (fdlibm) but mp3 didn't seem to
like the numbers it gave back. I'm working on it.
- These changes are selected by the define AMIGAL

1.7 Speed

The file spot1.mp3 has been used for timing tests. It is a 10.9 second
stereo 44.1khz 16bit sample.

25Mhz030/881 decodes it in 12minutes15seconds.
[No other timings are available]

For the previous version of mpeg3play the timings were:
50Mhz060 decodes spot1.mp3 in 1min05seconds.
25Mhz030/881 decodes it in 17:05
40Mhz040 (WE) decodes it in 1:33

[NB the 060 would definitely benefit from an 060 optimized compile]

1.8 Source

The source code to mpeg3play is available from
Johan Hagman's WWW.
My changes to this source are in the src directory of this archive ↩
. Just
copy my files over the ones in the distribution.

1.9 Masochists

For those of you with more CPU cycles than you know what to do with, I've included a first compile of an mp3 encoder. This is based on the source from
    ftp://ftp.tnt.uni-hannover.de/pub/MPEG/audio/mpeg2/software/dist10.tar.gz

This compile has no optimizations whatsoever, and could probably be a lot faster. I've only compiled it for 020/881 or 040/881 because you really wouldn't want to use anything that doesn't have an FPU on board.

A look at the timing. I tested with encoding the spot1.pcm file (the result of decoding the spot1.mp3 file). This is a 10.9 second 44khz 16bit stereo sound file. I was encoding with my 25Mhz 030/881. (As a ballpark figure, a 50Mhz 060 is about 20 times faster than a 25Mhz 030/881).

```
spot1.pcm -> spot1.mp2  192 kbps. psychoacoustic model 1.   35minutes
spot1.pcm -> spot1.mp3  192 kbps. PA model 2.               76 minutes
spot1.pcm -> spot1.mp3  128 kbps. PA model 2.               93 minutes
```

So encoding spot1.mp3 back to the 128kbps mp3 from which it came takes about 512 times as long as it takes for the sample to play. That's 26hours to encode a 3 minute song on a 25MHz030/881. With an 060 it's probably less than an hour and a half (guesstimate)

NB: mpeg audio encoders seem to only want 44khz 16bit stereo samples as input. Feeding it other spec sound samples results in gibberish in my (limited) experience.

1.10 Problems

- It's slow.
The mp3 algorithms are CPU intense. It's 5 times slower than realtime on a 50Mhz060, so don't expect miracles.

- Error: premature end of file (when playing back files with play16)
The sound actually sounds ok. I think it's got something to do with how mpeg3play decodes partial frames at the end of a sound file

- Immediate guru when run
Make sure you have the latest ixemul.library from aminet.
Or try the libnix versions.

- EMT trap
This will happen when you run a version of mpeg3play requiring an FPU but don't actually have an FPU on your machine. Run a version that doesn't require an FPU.
NB: the 020-040 version requires an FPU.

1.11 Feedback

```
                Feedback if you want to give it:

    - Do you care whether it requires ixemul.library or not?
    - Do you have any experience in putting ASM code into gcc code?
    - Any general suggestions?
    - Are you recoding the mp3 files into mp2 files directly, and hence
  using the '-r' switch?
    - How long does it take your computer to decode 'spot1.mp3'? File
  available from http://www.netforward.com/cryogen/?mikecheng
    - Which processor do you have and which one of the exe's do you use?

Contact
                me
```

1.12 Future

```
    - Merge my changes back with the author.
  Merged my ixemul version sources back. (not too many
  changes there) I'm not sure if
  i want to start merging back the libnix hacks. (all the
  libnix hacks are controlled by the switch AMIGAL )

    - Work out the problem with libnix compiling without an FPU.

    - compile for specific processors. (namely 060)
  Unfortunately gcc doesn't yet have an 060 optimization option.
  SAS/C does though, so someone may want to send me an 060 binary
  to include, or release it to aminet.

    - Replace some of the critical routines with ASM.
  Check the src for README.mpeg3play which has a profile
  of the code. 50% of the time is spent in SubBandSynthesis.
  A routine ripe for tweaking.
  Routines earmarked for optimization by the mpeg3play author
  are:    SubBandSynthesis()
    II_dequantize_sample()
    inv_mdct()
  [I have *NO* ASM talent whatsoever, so someone else is going to
   have to do this :)   ]
   Someone actually sent me some ASM code for subbandsynthesis
   but i have no idea how to put it into gcc. Help is welcome.

    - Do any obvious optimizations of the encoder.
```

1.13 Acknowledgements

```
    - Fraunhofer Institute for their generic C source
  (ftp:ftp.fhg.de/pub/layer3)
    - Johan Hagman for his optimized source code.
  Johan.Hagman@mailbox.swipnet.se
```

```
  http://home1.swipnet.se/%7Ew-10694/helpers.html
    - Troels for incorporating his amiga changes back into the source
  Troels Walsted Hansen <troels@stud.cs.uit.no>
```

1.14 Author

```
                I only did the amiga compile of the source. Contact me for amiga  ↩
                   specific
problems.

See
                 Acknowledgements
                 for a list of the real workers.

Michael Cheng
mikecheng@cryogen.com
http://www.netforward.com/cryogen/?mikecheng
Cstar on #amiga

Feel free to mail me....
```