TEX 3.1 and METAFONT 2.7 for the Atari ST

README for the Executables
9 September 2022

by Frank Ridderbusch

Preface

If you have already read earlier versions of this readme file, you can directly turn to the section `"Dynamic memory allocation"`.

There is one important modification, I made, that I would like to mention just at the beginning. There are **no** files `initex.ttp` or `inimf.ttp` any longer, as there have been in earlier versions. I have merged the functions which are unique to `initex.ttp` or `inimf.ttp` into one executable (no big deal, only changing some defines). Therefore only the files `tex.ttp` and `mf.ttp` are in the archive. The specific '`initex`' or '`inimf`' functions (dumping) are selected by a commandline option. (See section '`Installation and Usage`')

General Remarks

This archive contains executables for the Atari ST. All executables were generated from the WEB and the WEB2C sources version 5.8a, which Karl Berry announced and released in a TeXhax digest, which appeared on my news site during November 1990.

The Atari executables were first compiled with the GNU C compiler v1.37.1 running in an Un*x cross-compiling environment. The resulting objects were then linked with the C-library (at patchlevel 58), which J.R.Bammi and E.R.Smith put together (Thanks to both of them for their good work). Later, after some additional minor changes, I recompiled the C sources with GCC v1.38 and maximum optimization at home on my ST, which is equipped with 2,5 meg. After obtaining GCC v1.40, the final recompilation took place with this version. The executables are currently linked with the library (version 8) from E.R.Smith, which he build for MiNT, his multitasking TOS extension. Additionally the archive contains a BigTeX version with very large buffers (for those of you, who have a Mega 4 ST and use very large macro packages).

If you don't have a cross compiling environment and want to recompile the sources, you can just as well convert the WEB sources to C on your local Un*x box and then carry the C sources home. If you want to do this, it a good idea to decrease the constant `MAXLINES` to '`1000`' in the file `src-5.8a/web2c/splitup.c`. After that, you should be able to compile all files.

Implementation Details

The files from the WEB2C kit required only very minor changes, mostly things concerning the slash as a path separator opposed to the backslash the Atari uses. TeX and METAFONT both discard any '`\r`' (Carriage Return) character before a '`\n`' (Newline) in the input of normal tex files (The everlasting difference between Un*x and other systems; '`\n`' versus '`\r\n`' as line separators). I also modified the programs `pltopf` and `vptovf` to ignore any '`\r`' characters. If you encounter any problems with error messages about illegal character in the input file, try at first to remove any '`\r`' characters before you investigate further. It should be trivial to write a small utility, which removes all '`\r`' characters from an input file.

TeX and METAFONT both passed the trip and the trap tests on the Atari.

METAFONT doesn't have any window support on the ST. (If someone would write a window interface for the ST, I would be very interested.) I don't mean a GEM interface with this remark. METAFONT has a rudimentary window interface, to display proof mode character on screen. The WEB2C kit provides support for X11, X10 and some others.

All executables will open any file in binary mode.

Installation and Usage

The complete TEX and METAFONT environment is controlled via environment variables ala Un*x. On my Atari I've set the following values (as an example under gulam):

```
# All TeX font metric files are here.
setenv TEXFONTS f:\tex\texinput\tfm
# The dumped format files are found here.
setenv TEXFORMATS f:\tex\formats
# search paths for TeX input files
setenv TEXINPUTS .;f:\tex\texinput\latex\styles;f:\tex\texinput\macros
# The TeX string pool file is found here.
setenv TEXPOOL f:\tex
# The Metafont string pool file is found here.
setenv MFPOOL f:\metafont\mfbases
# The dumped base file is found here.
setenv MFBASES f:\metafont\mfbases
# search paths for Metafont input files
setenv MFINPUTS .;f:\metafont\mfinputs\cmr
```

(For a detailed description the above environment variables see the manual pages in the file `manpages.lzh`, also available from the atari archive at terminator.)

The executables don't care about slashes or backslashes as path separators. Both can be used. Don't mind, that the string pool files are called `*.poo` instead of `*.pool'`. The executables have no problems to find them.

The '`initex`' or '`inimf`' specific functions (dumping) are selected by giving the option `-i` as the last argument. Therefore to create the `plain.fmt` you have to issue the following commandline (assuming the environment variables are set correctly):

```
tex 'plain \dump' -i
```

The same applies to METAFONT. As a sidenote, the `plain.fmt` file from the BigTEX is about 110 KB larger than the one from the normal TEX, which is about 168 KB in size. The absolute sizes vary depending on the sizes of the internal buffers.

Also the utility programs may search along some path for some files set by an environment variable. So, if one program complains about a file, that it can't find although it is in the current directory, prefix the filename with '`./`' ot '`.\`'.

Dynamic Memory allocation for TEX

The TEX versions dated later than March 2nd, 1991 have a dynamic memory allocation scheme for most of the large arrays. The startup message reflects this (TeX Version 3.1t2var). There are certainly better schemes, to achieve dynamic allocation, but I found, that the chosen approach required only very minimal changes to the source files.

Originally the header file `texd.h` (part of the sources) contained some defines for the values of `memmax`, `memtop`, `triesize` etc. I changed these defines into variables and also changed the dependent array declarations into pointer declarations. Now, before the main TEX code gets control, these pointer are initialized with malloced memory. Since I don't know, if there are any places in the code, which assume that uninitialized memory is cleared, the flag, which prevents TOS from clearing the whole TPA should not be set. This is to make really sure, that malloced memory is cleared. If you have a TOS version earlier than 1.4, you don't have to bother with this, since these version clear the whole memory with every program launch.

To be able to modify the size of the arrays in the executables, I borrowed the idea of the programs `fixstk.ttp` and `printstk.ttp`, which come with the binary distribution of GNU-C for the ST. These two programs allow the modification or the display of the variable `stksize`, which determines, how the stack and the memory is used by a program.

TEX-CONF.TTP

I therefore wrote a program called `tex-conf.ttp`, which allows the modification and the display of TEX's main configuration values. This works only if the executable is unstripped, because the information, which is present in the symbol table is used to locate the variables in the executable. Therefore you should never strip the TEX or METAFONT executables, or, if you do, then keep unstriped versions around.

A sample output of `tex-conf.ttp` is shown in the following lines, if it is invoked without any options (You also see the default configuration for the files as they are in the archive).

```
f:\tex >> tex-conf bigtex.ttp tex.ttp mf.ttp

TEX-CONF: The Configurator for TeX and METAFONT by fgth (Jul 18 1991)

bigtex.ttp: TEXT segment is 190556 (187K)
bigtex.ttp: DATA segment is 1084 (2K)
bigtex.ttp: BSS segment is 239258 (234K)

bigtex.ttp: bufsize is 3000 (3K)
bigtex.ttp: dvibufsize is 16384 (16K)
bigtex.ttp: fontmemsize is 50000 (196K)
bigtex.ttp: maxstrings is 7500 (30K)
bigtex.ttp: memmax is 131070 (512K)
bigtex.ttp: memtop is 131070 (0K)
bigtex.ttp: poolsize is 100000 (98K)
bigtex.ttp: savesize is 4000 (16K)
bigtex.ttp: stksize is 16384 (0K)
bigtex.ttp: triehash is 16000 (219K)
bigtex.ttp: triesize is 16000 (79K)

bigtex.ttp: approx. memory usage running as INITEX 1624562 (1587K)
bigtex.ttp: approx. memory usage running as VIRTEX 1400562 (1368K)

tex.ttp: TEXT segment is 197132 (193K)
tex.ttp: DATA segment is 1084 (2K)
tex.ttp: BSS segment is 69986 (69K)
tex.ttp: bufsize is 500 (1K)
tex.ttp: dvibufsize is 4096 (4K)
tex.ttp: fontmemsize is 30000 (118K)
tex.ttp: maxstrings is 5000 (20K)
tex.ttp: memmax is 50000 (196K)
tex.ttp: memtop is 50000 (0K)
tex.ttp: poolsize is 45000 (44K)
tex.ttp: savesize is 1000 (4K)
tex.ttp: stksize is 16384 (0K)
tex.ttp: triehash is 12000 (165K)
tex.ttp: triesize is 12000 (59K)

tex.ttp: approx. memory usage running as INITEX 889798 (869K)
tex.ttp: approx. memory usage running as VIRTEX 721798 (705K)

mf.ttp: TEXT segment is 198472 (194K)
mf.ttp: DATA segment is 1112 (2K)
mf.ttp: BSS segment is 121894 (120K)

mf.ttp: bufsize is 500 (1K)
mf.ttp: gfbufsize is 16384 (16K)
mf.ttp: maxstrings is 2000 (10K)
mf.ttp: memmax is 45000 (176K)
mf.ttp: memtop is 45000 (0K)
mf.ttp: poolsize is 32000 (32K)
mf.ttp: stksize is 16384 (0K)

mf.ttp: approx. memory usage for METAFONT 560362 (548K)
```

First of all, the program prints the sizes of the TEXT, DATA and BSS segments of the executable. These values are found in the header of the program file. Then the current values of some variables are printed. The number of these variables differ depending on the file, which is currently examined. Most of these variables define the size of a memory array.

The variable `stksize` has the same meaning as in other GNU-C produced executables. It determines the usage of stack and memory. '`16k`' means, that the executable will run on a 16 KB stack. This size works okay for me, but if you experience strange effect (bombs) while running TeX you might increase the value. Previously I had set `stksize` to '`-1L`', which would cause the program to grab every bit of memory it could get and do mallocs from internal heap. But that's not very nice in a multitasking environment (aka MiNT).

The variable `memtop` doesn't actually define a memory array, but it is presented, since it's value must be equal to `memmax`, when the `tex.ttp` runs as INITeX (`-i` option). `memmax` may be larger than `memtop` for normal operation, but both (`memtop` and `memmax`) may not be larger than 65530 for a normal sized TeX or METAFONT and not larger than 262140 for a BigTeX. The variable `triehash` is for information purposes only. The printed value is directly dependent on `triesize` and is only used when the memory usage for INITEX is calculated. INITEX uses 6 additional arrays for the handling of hyphenation pattern, which are all subsumed under the name `triehash`. The 6 additional arrays are also the reason, why an INITEX with static arrays is so much bigger than VIRTEX with the same sizes.

The other values directly define the size on an array. The values in braces show the actual memory, which is allocated for that array. The last line, that is printed, is the calculated memory usage, which is basically the sum of all shown arrays plus the TEXT-, DATA- and BSS segments sizes.

Configuring TEX and METAFONT

Now to change a value, you simple give the value name as an option during program invocation. For example:

```
tex-conf -bufsize 2000 h:\tex\initex.ttp h:\tex\virtex.ttp
```

or

```
tex-conf -memtop 50000 -memmax 50000 h:\tex\initex.ttp \
    h:\tex\virtex.ttp
```

Changing a value makes it in most cases necessary to create new dumped formats with '`tex -i`' (*.FMT files).

I made all the above mentioned changes in the first place to get TeX also running on a 1 meg ST. For a 1 meg ST you might modify `dvibufsize`, '`memmax = memtop`', and `triesize`. I was able to create *.FMT formats on a 1 meg ST (artificially achieved with a 1536 kb ram disk on my 2,5 meg ST) with '`memmax = memtop = 50000`', '`triesize = 12000`' and '`dvibufsize = 4096`', but I must admit there was nothing in memory except the hard disk driver and an environment setting program. A `triesize` of 12000 is necessary for german hyphenation pattern. If you only use English hyphenation pattern you can decrease `triesize` to 8000. This in turn might allow you to increase '`memmax = memtop`' to maximum. Remember, the main problem was initex with its 6 additional array for hyphenation compared with virtex. The increase of `triesize` by 1000 costs 15000 bytes of memory.

I've included the program `setenv.prg` and a file `env.inf` in the TEX executable archive. '`setenv.prg`' is an environment setter for the AES. '`setenv.prg`' goes into the auto folder and '`env.inf`' into the root directory of your boot drive. '`env.inf`' is included as an example for a possible configuration. I've included this program, to allow the invocation of TEX from the desktop.

Concluding this section, here is a short recipe to produce `plain.fmt` files on a 1 meg ST.

- check, that '`memmax = memtop = 50000`', '`triesize = 12000`' and '`dvibufsize = 4096`'. (This is done with `tex-conf.ttp` running from a CLI)
- edit '`env.inf`' to reflect your current setting.
- copy '`env.prg`' to the auto folder and '`env.inf`' in root directory of your boot drive.
- strip the auto folder to the bare minimum to get maximum memory (only your hard disk driver and '`env.prg`' should stay in place)
- reboot your ST to activate the new auto folder setting.

- invoke '`tex.ttp`' from the desktop with the arguments '`plain \dump -i`' or, if you start `tex.ttp` with only the '`-i`' option, type '`plain \dump`', when TEX prompts with '`**`' (This creates the `plain.fmt` file)
- repeat the above line if necessary for LaTEX.
- reboot the ST with your old configuration.
- Now you can now happily invoke '`tex.ttp`' from a CLI and become a TEX wizard.

WARNING

The complete port is sort of a quick and dirty port. I'm quite happy with the way it works, since I prefer a CLI environment like gulam. I'm also not a TEX or METAFONT wizard. I only compiled the sources. So, if you experience problems or have questions, which concerns TEX's or METAFONT's inner workings, DON'T write to me. I won't be able to help you. Instead post an article in the news group '`comp.text.tex`'.

If you have Atari ST specific questions about this implementation, you CAN write to me.

If you are interested, I can also send a file containing the diffs relative to the files from Karl Berry (the original WEB2C kit), but I won't send the complete WEB and WEB2C sources.

Additionally you will need the latest versions of the various TEX and Metafont macro packages like Plain, LaTEX etc. Good places to look for these are the following archive sites:

- [archive-server@]sun.soe.clarkson.edu: Email and anon ftp
- [mail-server@]cs.ruu.nl: Email and anon ftp (for European users)
- labrea.stanford.edu: anon ftp (The official TEX archive site)
- [mailserv@]ymir.claremont.edu: Email and anon ftp
- [mailserv@]rusmv1.rus.uni-stuttgart.de: Email and anon ftp (for German users)

Unsorted List of Changes

The following is a list of differences between this TEX and METAFONT executables and the ones, which were available from the Atari archive at terminator during the end of 1990 until the middle of Jan 1991.

- Changed the path separator from '`:`' to '`;`'. This change allows the use of standard TOS path names (ex. '`f:\tex`'). Prior to this you had to use the '`/dev/<partition>`' notation for path names. All the utility programs still use this notation, since I didn't rebuild them with the modified support files in the common directory.
- Subdirectory search is enabled.
- Upon startup a message is printed, when the executable was compiled.
- Compiled with GNU-C v1.39 and maximum optimization (`-O`, `-fomit-frame-pointer`, `-fcombine-regs`. The additional flag `-fstrength-reduce` caused errors in the DVI file.)
- Instead of ignoring all '`\r`' characters in the input stream, only the '`\r`' from a '`\r\n`' pair is removed.

Change to the version compiled later than March 2, 1991

- Dynamic memory allocation for most of the internal arrays.
- Startup modified to reflect the dynamic allocation.

Contacting me

You can contact either by electronic or snail mail.

```
Email:
ridderbusch.kd@sni-usa.com (America (North & South))
ridderbusch.kd@sni.de (Rest of World)

Snail:
Frank Ridderbusch
Sander Str. 17
W-4790 Paderborn
Germany
```