tail-f.exe: Simple tail for Microsloth Windows

Gene Michael Stover

created 2004 February
updated Sunday, 2006 March 12

Contents

1 What is this?

Sometimes you want to run "`tail -f myfile`", but you are in a Microsloth Winders environment. Looks like a job for Cygwin, but maybe you don't have access to Cygwin, you don't have permission to install software on the computer, or the amount of time you'll be using the computer (if you can run that command) is so short that it doesn't justify the effort of installing Cygwin.

You need `tail-f.exe`. It does what "`tail -f`" does on unix.[1]

---

[1] If you think unix is an implementation, then allow me this errant belief, & when I write "unix", you read "unix-like operating system".

2 License

The license agreement is the "MIT license". It is in the header of the source code file.

3 Obtaining

The single source file is in Appendix B.

The single executable file is `http://cybertiggyr.com/gene/tlf/tail-f.exe`.

4 Compiling

The executable file should be fine, but if you want to compile it yourself, you should be able to do it with a single compile command. The program is self-contained & in just one file.

I think later versions of Visual C++ will bitch about the old-style function argument definitions I use. Too bad for Visual C++.

5 Installing

To install, just copy the `tail-f.exe` file into your file system. For ease of use, I recommend a location which is in your path. I often use `C:\bin\`.

There is no installation program. There are no Registry entries or support files to create or edit.

To un-instal, just remove the `tail-f.exe` file from your file system.

6 Using

Assuming `tail-f.exe` is in your path, to use `tail-f.exe` on a file called *debug.log*, run this command in a "`cmd`" window: "`tail-f debug.log`". This will behave as "`tail -f debug.log`" would on unix.

To end a `tail-f.exe` process, type Ctrl-C into the terminal window.

There are no command line options, & `tail-f.exe` does not read from standard input. The only way to use it is to give it exactly one filename on the command line.

7 Limitations

The real `tail` command on unix supports many options. `tail-f.exe` supports none of them other than "`-f`".

Notice that you do not give a "`-f`" option to `tail-f.exe`. The "`-f`" is part of the program's name.

A Other File Formats

- This document is available in multi-file HTML format at http://cybertiggyr.com/gene/tlf/.

- This document is available in Pointless Document Format (PDF) at http://cybertiggyr.com/gene/tlf/tlf.pdf.

B Source Code (`tail-f.c`)

This source code is also online at `http://cybertiggyr.com/gene/tlf/tail-f.c`.

```
/*
 * $Header: /home/gene/library/website/docsrc/tlf/RCS/tail-f.c,v 395.1 2008/04/20 17:25:48 gene Exp $
 *
 * Copyright (c) 2004--2006 Gene Michael Stover.  All rights reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL GENE MICHAEL STOVER BE LIABLE FOR
 * ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
 * WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 *
 * Except as contained in this notice, the name of Gene Michael Stover
 * shall not be used in advertising or otherwise to promote the sale, use
 * or other dealings in this Software without prior written authorization
 * from Gene Michael Stover.
 */

/*
 * A simple re-implementation of the "tail" program found on unices.
 * It's for when you are stuck on a computer without a tail.
 */

#define MICROSLOTH_WINDERS 1
```

```c
/* Standard C */
#include <stdio.h>
#include <stdlib.h>

/* Proprietary bullshit */
#ifdef MICROSLOTH_WINDERS
#include <windows.h>
#endif

typedef int Boolean;

static void
S_Sleep (n)
     int n; /* seconds */
{
#ifdef MICROSLOTH_WINDERS
  /* Dumbshit Microsloth Winders doesn't have a "sleep" function.  God
   * forbid Microsloth include an ANSI/ISO Standard C function in
   * their library.  Proprietary idiot fuckers like Microsloth are what's
   * wrong with capitalism. */
  Sleep (n * 1000);
#else
  sleep (n);
#endif
}

static Boolean
S_IsReadable (pathname)
     char pathname[];
{
  FILE *fp;

  fp = fopen (pathname, "r");
  if (fp != NULL) {
    fclose (fp);
  }
  return fp != NULL;
}

static long
S_InitOffset (pathname)
     char pathname[];
{
  long offset;
  FILE *fp;

  fp = fopen (pathname, "r");
  if (fp != NULL) {
    if (fseek (fp, 0, SEEK_END) == 0) {
      offset = ftell (fp);
```

4

```
        offset -= 1024;                    /* backup 1 kilobyte */
        if (offset < 0) {
          offset = 0;
        }
      } else {
        /* Couldn't seek to the end. */
        offset = -1;
      }
      fclose (fp);
    } else {
      /* Couldn't open the file to read. */
      offset = -1;
    }
    return offset;
}

static long
S_Tick (pathname, offset)
     char pathname[];
     long offset;
{
  FILE *fp;
  int c;

  fp = fopen (pathname, "r");
  if (fp != NULL) {
    if (fseek (fp, offset, SEEK_SET) == 0) {
      while ((c = fgetc (fp)) != -1) {
        putchar (c);
      }
      offset = ftell (fp);
    } else {
      fprintf (stderr, "\n%s:%d: Can't seek to offset %ld.\n", __FILE__,
               __LINE__, offset);
      offset = -1;
    }
    fclose (fp);
  } else {
    fprintf (stderr, "\n%s:%d: Can't re-open the file.  Did it move?\n",
             __FILE__, __LINE__);
    offset = -1;
  }
  return offset;
}

int
main (argc, argv)
     int argc;
     char *argv[];
{
```

```
    int rc = 0;
    char *pathname;
    long offset;

    setbuf (stdout, NULL);
    pathname = argv[1];
    if (S_IsReadable (pathname)) {
      offset = S_InitOffset (pathname);
      while (rc == 0 && offset >= 0) {
        offset = S_Tick (pathname, offset);
        if (offset >= 0) {
          S_Sleep (2);
        } else {
          fprintf (stderr, "%s:%d: S_Tick indicates error\n", __FILE__,
                   __LINE__);
          rc = -7;
        }
      }
    } else {
      fprintf (stderr, "%s:%d: It appears that \"%s\" does not exist.\n",
               __FILE__, __LINE__, pathname);
      rc = -4;
    }
    return rc == 0 ? 0 : 1;
}

/* --- end of file --- */
```

References