Writing NROFF Terminal Descriptions

Eric Allman
Britton-Lee, Inc.

1. INTRODUCTION

As of the Version 7 Phototypesetter release of UNIX,* NROFF has supported terminal description files. These files describe the characteristics of available hard-copy printers. This document describes some of the details of how to write terminal description files.

Disclaimer. This document describes the results of my personal experience. The effects of changing some of the fields from the norms may not be well defined, even if it seems like it ‘‘ought’’ to work given the descriptions herein. These tables are known to vary slightly for different versions of UNIX. I have not seen UNIX 3.0 at this time, so this may be irrelevant in that context.

2. GENERAL

When NROFF starts up, it looks for a −**T** flag describing the terminal type. For example, if the command line is given as

nroff −T300s

NROFF prepares output for a *DTC300S* terminal. This terminal is described in the file /usr/lib/term/tab300s on most systems.

If no −**T** flag is given, the terminal type **37** (ASR 37 − a relic assumed for historical humor only) is assumed.

The terminal description table is a stripped ‘‘.o’’ file generated from a data structure, shown in figure one. This structure can be dealt with in two sections: the terminal capability descriptor (everything to codetab), and the output descriptor.

3. TERMINAL CAPABILITIES

The section of the data structure up to but excluding *codetab* describes the basic functions and setup requirements of the terminal. Distances are measured in ‘‘units,’’ which are 1/240 of an inch in NROFF. In general, NROFF assumes that there is a ‘‘plot mode’’ on the terminal that allows you to move in small increments. A terminal has a resolution when in plot mode that is measured in units. This limits how well the terminal can simulate printing Greek and special characters.

3.1. bset, breset

These fields define bits in a vanilla stty(2) word (sg_flags) to set and clear respectively when NROFF starts. They are normally represented in octal, although you could include <sgtty.h>. [Note: these fields are presumably different in UNIX 3.0.]

3.2. Hor, Vert

These represent the horizontal and vertical resolution respectively of the terminal when it is in plot mode. They are given in units.

---

*UNIX is a trademark of Bell Laboratories.

```
#define INCH 240           /* one inch in units */
struct
{
        int bset;          /* stty bits to set */
        int breset;        /* stty bits to reset */
        int Hor;           /* horizontal resolution in units */
        int Vert;          /* vertical resolution in units */
        int Newline;       /* the distance a newline moves */
        int Char;          /* the distance one char moves */
        int Em;            /* size of an Em */
        int Halfline;      /* the distance a halfline up/down moves */
        int Adj;           /* default adjustment width */
        char *twinit;      /* string to init the terminal */
        char *twrest;      /* string to reset the terminal */
        char *twnl;        /* string to send a newline (CR-LF) */
        char *hlr;         /* half line reverse string */
        char *hlf;         /* half line forward string */
        char *flr;         /* full line reverse string */
        char *bdon;        /* string to turn boldface on */
        char *bdoff;       /* string to turn boldface off */
        char *ploton;      /* string to turn plot on */
        char *plotoff;     /* string to turn plot off */
        char *up;          /* move up in plot mode */
        char *down;        /* move down in plot mode */
        char *right;       /* move right in plot mode */
        char *left;        /* move left in plot mode */
        char *codetab[256-32];  /* the codes to send for characters */
        int zzz;           /* padding */
};
```

Figure 1 – the terminal descriptor data structure

3.3. Newline

This field describes the distance that the *twnl* field (below) will move the paper; it is literally the size of a newline.

3.4. Char

This is the distance that a regular character will move the print head to the right.

3.5. Em

The ‘‘em’’ is a typesetting unit, approximately equal to the width of the letter ‘‘m’’. In NROFF driver tables, this must be the distance a space or backspace character will move the carriage.

3.6. Halfline

This is the distance that the *hlr* or *hlf* strings move the print head (reverse or forward respectively).

3.7. Adj

This is the resolution that NROFF will normally adjust your lines to horizontally. Typically this is the same as Char. If the **–e** flag is given to NROFF, output resolution will be to the full device resolution.

3.8. twinit, twrest

These strings are output when NROFF starts and finishes respectively.

3.9. twnl

This string is output when NROFF wants to do a carriage return. Typically it will be ‘‘\r\n’’. Remember, the terminal will normally have CRMOD turned off when this is set.

3.10. hlr, hlf

These strings are sent to move the carriage back or forward one half line respectively. The actual amount that they moved is defined by Halfline. The carriage should be left in the same column.

3.11. flr

The string to send to move a full line backwards. This should leave the carriage in the same column.

3.12. bdon, bdoff

These strings are sent to turn boldface mode on and off respectively. Normally this will set the terminal into overstrike mode. If they are not given, some newer versions of NROFF will output the characters four times to force overstriking.

3.13. ploton, plotoff

These strings turn plot mode on and off respectively. In plot mode, the carriage moves a very small amount, and only under specific control; i.e., characters do not automatically cause any carriage motion.

3.14. up, down, right, left

These strings are only output in plot mode. They should move the carriage up, down, left, and right respectively; they will move the carriage a distance of Hor or Vert as appropriate.

3.15. An Example

Consider the following table describing a DTC300S:

```
/*bset*/          0,
/*breset*/        0177420,
/*Hor*/           INCH/60,
/*Vert*/          INCH/48,
/*Newline*/       INCH/6,
/*Char*/          INCH/10,
/*Em*/            INCH/10,
/*Halfline*/      INCH/12,
/*Adj*/           INCH/10,
/*twinit*/        "\033\006",
/*twrest*/        "\033\006",
/*twnl*/          "\015\n",
/*hlr*/           "\033H",
/*hlf*/           "\033h",
/*flr*/           "\032",
/*bdon*/          "",
/*bdoff*/         "",
/*ploton*/        "\006",
/*plotoff*/       "\033\006",
/*up*/            "\032",
/*down*/          "\n",
/*right*/         " ",
/*left*/          "\b",
```

This describes a terminal that should have the ALLDELAY and CRMOD bits turned off, 1/60" horizontal and 1/48" vertical resolution, six lines per inch and ten characters per inch, including space, halfline takes 1/12" (one half of a full line), should send ESC-control-F to initialize and reset the terminal (to insure that it is in a normal state), takes <CR><LF> to give a newline, <ESC>H to move back one half line, <ESC>h to move forward one half line, control-Z to move back one full line, has no bold mode, takes control-F to enter plot mode and escape-control-F to exit plot mode, and uses control-Z, linefeed, space, and backspace to move up, down, right, and left respectively when in plot mode.

4. CHARACTER DESCRIPTIONS

There is one character description for each possible character to be output. The easiest way to find what character corresponds to what position is to edit an existing character table; one is given in the appendix as an example. Character representations are represented as a string per character.

The first character of the string is interpreted as a binary number giving the number of character spaces taken up by this character. For regular characters this will always be ‘‘\001’’, but Greek and special characters can take more. If the 0200 bit is set in this character, it indicates that the character should be underlined if we are in italic (underline) mode. Thus, alphabetic and numeric descriptions will begin ‘‘\201’’.

The remainder of the string is output to represent the character. If the first output character (i.e., the second character in the total string) has the 0200 bit set, the character will be output in plot mode so that fancy characters can be built up from existing characters. If necessary, the ‘‘\200’’ character can be used as a null character to force NROFF to set the terminal into plot mode. All characters without the 0200 bit are output literally; characters with the 0200 bit are not output, but are used to indicate local carriage movement. The next two bits (0140 bits) represent direction:

0200 right
0240 left
0300 down
0340 up

The bottom five bits represent a distance in terminal resolution units. This is rather confusing, but the examples should make this much more clear.

4.1. Some Examples

The following examples are from the DTC300S table:

```
"\001 ",          /*space*/
"\001=",          /*=*/
"\201A",          /*A*/
```

These entries show that all of these characters take one character width when output. The letter A is underlined in italic mode, but neither space nor equal sign is.

```
"\001o\b+",       /*bullet*/
"\002[]",         /*square*/
"\202fi",         /*fi*/
```

The bullet character takes only one character position, but is created by outputing the letter ‘‘o’’ and overstriking it with a plus sign. The square character is approximated with two brackets; it takes two full character positions when output. The ‘‘fi’’ ligature is produced using the letters ‘‘f’’ and ‘‘i’’ (surprise!); it is underlined in italic mode.

```
"\001\241c\202(\241", /*alpha*/
"\001\200B\242\302|\202\342", /*beta*/
```

The letters alpha and beta both take a single character position. The alpha is output by entering plot mode, moving left 1 terminal unit (1/60" if you recall), outputing the letter ‘‘c’’, moving right 2/60", outputing a left parenthesis, and finally moving left 1/60"; it is critical that the net space moved be zero both horizontally and vertically. The beta first has a dummy 0200 character to enter plot mode but not output anything. It then outputs a ‘‘B’’, moves left 2/60", moves down 2/48", outputs a vertical bar (which is designed to partically overstrike the left edge of the ‘‘B’’, and finally move right 2/60" and up 2/48" to set us back to the right place.

5. INSTALLATION

To install a terminal descriptor, make it up by editing an existing terminal descriptor. Assuming your terminal name is *term*, call your new descriptor **tab***term***.c**. Then, execute the following commands:

```
cc -c tabterm.c
strip tabterm.c
cp tabterm.o /usr/lib/term/tabterm
```

The directory /usr/src/cmd/troff/term typically has a shell file to do this.

APPENDIX

A Sample Table

This table describes the DTC 300S.

```
#define INCH 240
/*
DASI300S
nroff driving tables
width and code tables
*/

struct {
        int bset;
        int breset;
        int Hor;
        int Vert;
        int Newline;
        int Char;
        int Em;
        int Halfline;
        int Adj;
        char *twinit;
        char *twrest;
        char *twnl;
        char *hlr;
        char *hlf;
        char *flr;
        char *bdon;
        char *bdoff;
        char *ploton;
        char *plotoff;
        char *up;
        char *down;
        char *right;
        char *left;
        char *codetab[256-32];
        int zzz;
        } t = {
/*bset*/  0,
/*breset*/        0177420,
/*Hor*/           INCH/60,
/*Vert*/ INCH/48,
/*Newline*/       INCH/6,
/*Char*/INCH/10,
/*Em*/            INCH/10,
/*Halfline*/      INCH/12,
/*Adj*/           INCH/10,
/*twinit*/        "\033\006",
/*twrest*/        "\033\006",
/*twnl*/ "\015\n",
```

```
/*hlr*/          "\033H",
/*hlf*/          "\033h",
/*flr*/          "\032",
/*bdon*/"",
/*bdoff*/        "",
/*ploton*/       "\006",
/*plotoff*/      "\033\006",
/*up*/           "\032",
/*down*/         "\n",
/*right*/ " ",
/*left*/  "\b",
        /*codetab*/
"\001 ",  /*space*/
"\001!",  /*!*/
"\001\"",/*"*/
"\001#", /*#*/
"\001$", /*$*/
"\001%",         /*%*/
"\001&",         /*&*/
"\001’",  /*’ close*/
"\001(",  /*(*/
"\001)",  /*)*/
"\001*", /***/
"\001+", /*+*/
"\001,",  /*,*/
"\001-",  /*- hyphen*/
"\001.",  /*.*/
"\001/",  /*/*/
"\2010", /*0*/
"\2011", /*1*/
"\2012", /*2*/
"\2013", /*3*/
"\2014", /*4*/
"\2015", /*5*/
"\2016", /*6*/
"\2017", /*7*/
"\2018", /*8*/
"\2019", /*9*/
"\001:",  /*:*/
"\001;",  /*;*/
"\001<", /*<*/
"\001=", /*=*/
"\001>", /*>*/
"\001?", /*?*/
"\001@",         /*@*/
"\201A",/*A*/
"\201B",/*B*/
"\201C",/*C*/
"\201D",/*D*/
"\201E", /*E*/
"\201F", /*F*/
"\201G",/*G*/
"\201H",/*H*/
"\201I",  /*I*/
"\201J",  /*J*/
"\201K",/*K*/
"\201L", /*L*/
```

```
"\201M",		/*M*/
"\201N",/*N*/
"\201O",/*O*/
"\201P", /*P*/
"\201Q",/*Q*/
"\201R",/*R*/
"\201S", /*S*/
"\201T", /*T*/
"\201U",/*U*/
"\201V",/*V*/
"\201W",		/*W*/
"\201X",/*X*/
"\201Y",/*Y*/
"\201Z", /*Z*/
"\001[",  /*[*/
"\001\\", /*\*/
"\001]",  /*]*/
"\001^",  /*^*/
"\001_", /*_ dash*/
"\001‘",  /*‘ open*/
"\201a", /*a*/
"\201b", /*b*/
"\201c", /*c*/
"\201d", /*d*/
"\201e", /*e*/
"\201f",  /*f*/
"\201g", /*g*/
"\201h", /*h*/
"\201i",  /*i*/
"\201j",  /*j*/
"\201k", /*k*/
"\201l",  /*l*/
"\201m",		/*m*/
"\201n", /*n*/
"\201o", /*o*/
"\201p", /*p*/
"\201q", /*q*/
"\201r",  /*r*/
"\201s",  /*s*/
"\201t",  /*t*/
"\201u", /*u*/
"\201v", /*v*/
"\201w",/*w*/
"\201x", /*x*/
"\201y", /*y*/
"\201z", /*z*/
"\001{", /*{*/
"\001|",  /*|*/
"\001}", /*}*/
"\001~",  /*~*/
"\000\0",		/*narrow sp*/
"\001-",   /*hyphen*/
"\001o\b+",		/*bullet*/
"\002[]",  /*square*/
"\001-",   /*3/4 em*/
"\001_",  /*rule*/
"\000\0",		/*1/4*/
```

```
"\000\0",       /*1/2*/
"\000\0",       /*3/4*/
"\001-",  /*minus*/
"\202fi", /*fi*/
"\202fl", /*fl*/
"\202ff",       /*ff*/
"\203ffi",      /*ffi*/
"\203ffl",      /*ffl*/
"\000\0",       /*degree*/
"\000\0",       /*dagger*/
"\000\0",       /*section*/
"\001’",  /*foot mark*/
"\001’",  /*acute accent*/
"\001‘",  /*grave accent*/
"\001_",  /*underrule*/
"\001/",  /*slash (longer)*/
"\000\0",       /*half narrow space*/
"\001 ",  /*unpaddable space*/
"\001\241c\202(\241", /*alpha*/
"\001\200B\242\302|\202\342", /*beta*/
"\001\200)\201/\241", /*gamma*/
"\001\200o\342<\302", /*delta*/
"\001<\b-", /*epsilon*/
"\001\200c\201\301,\241\343<\302", /*zeta*/
"\001\200n\202\302|\242\342", /*eta*/
"\001O\b-", /*theta*/
"\001i",  /*iota*/
"\001k",  /*kappa*/
"\001\200\\\304\241’\301\241’\345\202", /*lambda*/
"\001\200u\242,\202", /*mu*/
"\001\241(\203/\242", /*nu*/
"\001\200c\201\301,\241\343c\241\301‘\201\301", /*xi*/
"\001o",  /*omicron*/
"\001\341-\303\"\301\"\343", /*pi*/
"\001\200o\242\302|\342\202", /*rho*/
"\001\200o\301\202~\341\242", /*sigma*/
"\001\200t\301\202~\243~\201\341", /*tau*/
"\001v",  /*upsilon*/
"\001o\b/", /*phi*/
"\001x",  /*chi*/
"\001\200/-\302\202’\244’\202\342", /*psi*/
"\001\241u\203u\242", /*omega*/
"\001\242|\202\343-\303\202‘\242", /*Gamma*/
"\001\242/\303-\204-\343\\\242", /*Delta*/
"\001O\b=", /*Theta*/
"\001\242/\204\\\242", /*Lambda*/
"\000\0",       /*Xi*/
"\001\242[]\204[]\242\343-\303", /*Pi*/
"\001\200>\302-\345-\303", /*Sigma*/
"\000\0",       /**/
"\001Y", /*Upsilon*/
"\001o\b[\b]", /*Phi*/
"\001\200[]-\302\202’\244‘\202\342", /*Psi*/
"\001\200O\302\241-\202-\241\342", /*Omega*/
"\000\0",       /*square root*/
"\000\0",       /*terminal sigma*/
"\000\0",       /*root en*/
```

```
"\001>\b_",		/*>=*/
"\001<\b_",		/*<=*/
"\001=\b_",		/*identically equal*/
"\001-",	/*equation minus*/
"\001=\b~",		/*approx =*/
"\000\0",		/*approximates*/
"\001=\b/",		/*not equal*/
"\002->",		/*right arrow*/
"\002<-",		/*left arrow*/
"\001|\b^",		/*up arrow*/
"\000\0",		/*down arrow*/
"\001=",	/*equation equal*/
"\001x",	/*multiply*/
"\001/",	/*divide*/
"\001+\b_",		/*plus-minus*/
"\001U", /*cup (union)*/
"\000\0",		/*cap (intersection)*/
"\000\0",		/*subset of*/
"\000\0",		/*superset of*/
"\000\0",		/*improper subset*/
"\000\0",		/* improper superset*/
"\002oo",		/*infinity*/
"\001\200o\201\301‘\241\341‘\241\341‘\201\301", /*partial derivative*/
"\001\242\\\343-\204-\303/\242", /*gradient*/
"\001\200-\202\341,\301\242", /*not*/
"\001\200|’\202‘\243\306’\241‘\202\346",	/*integral sign*/
"\000\0",		/*proportional to*/
"\000\0",		/*empty set*/
"\000\0",		/*member of*/
"\001+",	/*equation plus*/
"\001r\bO",		/*registered*/
"\001c\bO",		/*copyright*/
"\001|",	/*box rule */
"\001c\b/",		/*cent sign*/
"\000\0",		/*dbl dagger*/
"\000\0",		/*right hand*/
"\001*",	/*left hand*/
"\001*",	/*math * */
"\000\0",		/*bell system sign*/
"\001|",	/*or (was star)*/
"\001O", /*circle*/
"\001|",	/*left top (of big curly)*/
"\001|",	/*left bottom*/
"\001|",	/*right top*/
"\001|",	/*right bot*/
"\001|",	/*left center of big curly bracket*/
"\001|",	/*right center of big curly bracket*/
"\001|",	/*bold vertical*/
"\001|",	/*left floor (left bot of big sq bract)*/
"\001|",	/*right floor (rb of ")*/
"\001|",	/*left ceiling (lt of ")*/
"\001|"};/*right ceiling (rt of ")*/
```