

一种通用终端打印技术的设计与实现

石文昌 (广西计算中心)

在 UNIX / XENIX 的应用领域中,通过终端连接打印机来提供打印功能,已经是一种被广泛采用的方法。通常,对于某一厂家生产的某种具体的终端,利用终端提供的透明打印命令,编一个简单的程序,便可实现终端的打印输出。但是,在实际应用中,用户面对的是各种各样的终端,不同的终端提供不同的透明打印命令,如果为每种终端都写一个打印程序,那显然是极不方便的。本文介绍笔者设计的一种通用终端打印技术,它由一个 C 语言程序实现,可实现各种终端的打印输出,适用于任何运行 UNIX / XENIX 系统的机器。

终端打印程序的基本思想是:首先输出一个进入透明打印的命令,接着从标准输入读取信息,并把读到的信息原原本本地送到标准输出上,最后输出一个退出透明打印的命令。这里,借鉴了 vi 程序对不同类型的终端的处理策略,关键是把终端的透明打印命令与打印程序分离开来。在具体实施上,则把各种终端的透明打印命令写在一个称为命令定义文件的文本文件中,不妨把该文件命名为 prtermcap,该文件采用与文件 / etc / termcap 相同的格式,用户可以随时查阅或修改它。下面是文件 prtermcap 的格式及其说明的示例。终端打印由 C 语言程序 tm-print.c 实施。在处理技术上,程序借助环境变量沟通与命令定义文件之间的联系。具体地说,就是通过环境变量 TERMNAME 获取终端的名称,通过环境变量 PRTERMCAP 获取命令定义文件的全路径名,然后,以终端名为索引,在命令定义文件中检索出相应的透明打印命令,从而根据终端打印的基本思想实现透明打印。程序源码附后。

用以下命令编译源程序 tm-print.c 得到可执行程序 tm_print:

```
cc-o tm-print tm-print.c
```

把程序 tm_print 移到适当的目录中(如 /usr/bin 中),就可作为一个系统命令来使用了。要留意的是,使

用程序之前,需设置好环境变量,这可由以下形式的命令行来完成:

```
TERMNAME= 925a; PRTERMCAP= $
{HOME} / prtermcap; export TERMNAME
PRTERMCAP 此后,作为例子,可用以下命令在国光
CJ-925A 终端上打印出文件 file:
```

```
cat file|tm_print
```

程序 tm_print 不受任何终端限制,任何终端的打印命令都可由用户随时方便地填写到命令文件中,所以这一技术具有广泛的适用性,它已被用于我们开发的多个应用系统中,收到了很好的效果。

终端透明打印命令定义文件,以(##)开头的行是注释行

```
500|600| st500 |st600 |star500 |star600 |实达系列终端: \
: PN = \E[5i: PS = \E[4i:
925a|cj925a|国光 CJ-925A 终端: \
: PN = \E[5i: PS = \Ea:
```

每种终端由一个逻辑行定义。一个物理行的最后一个可显示字符

如果是反斜杠(\),则它与下临的那个物理行连成同一个逻辑行。

每个逻辑行分成若干记录,记录间由冒号(:)分隔,每个冒号标

志一个记录的结束。每行的第一个记录由竖线(|)分成若干项

, ## 每项登记一个终端名,最后一项可以是终端的有关说明。一个逻辑

辑行的其它记录描述的是第一个记录登记的终端所提供的命令。

PN = 进入透明打印的命令, PS = 退出透明打印的命令。 \E 表示

转义符(ESC)。

May 3 11:07 1993 Page 1

```

1 /* 通用终端打印程序, 适用于各种类型的终端 */
2 /* 需要设置两个环境变量: TERMNAME 和 PTERMCAP
3 * TERMNAME 是终端名, PTERMCAP 是透明打印命令说明文件路径名. PTERMCAP
4 * 文件的格式同 TERMCAP. PTERMCAP 中的 PN 项是进入透明打印的命令, PS 项
5 * 是退出透明打印的命令.
6 * \E -- <ESC>
7 */
8 # include <stdio.h>
9 # define ESC '\033'
10 # define NOTFOUND -1
11 # define SIZECMD 20 /* 透明打印命令宽度 */
12 # define SIZELINE 500 /* 命令定义文件行宽度 */
13 # define SIZETERM 14 /* 终端名宽度 */
14 # define SIZEFNAME 60 /* 命令定义文件路径名宽度 */
15 struct pcommand {
16     char PN[SIZECMD]; /* 进入透明打印命令 */
17     char PS[SIZECMD]; /* 退出透明打印命令 */
18 };
19 main()
20 {
21     char c;
22     struct pcommand cmdbuf;
23     getprcmd(&cmdbuf); /* 取透明打印命令 */
24     printf("%s", cmdbuf.PN); /* 进入透明打印 */
25     while ((c=getchar())!=EOF)
26         putchar(c);
27     printf("%s", cmdbuf.PS); /* 退出透明打印 */
28 } /* main() */
29 getprcmd(cmdbuf) /* 从命令定义文件中取透明打印命令 */
30 struct pcommand *cmdbuf;
31 {
32     FILE *fp;
33     char tname[SIZETERM], linebuf[SIZELINE];
34     strcpy(tname, getenv("TERMNAME")); /* 取终端名 */
35     if ((fp=fopen(getenv("PTERMCAP"), "r"))==NULL) /* 打开定义文件 */
36         goto nocmd;
37     while (getline(fp, linebuf)!=NOTFOUND) /* 取一定行 */
38         if (linebuf[0]!='#') /* 以 # 打头的行是注释行 */
39             if (matchtermname(tname, linebuf)) /* 如果终端名匹配 */
40                 getPNandPS(cmdbuf, linebuf); /* 从定义行中取透明打印命令 */
41                 fclose(fp);
42                 return;
43     }
44     nocmd:
45     fclose(fp);
46 }
47 cmdbuf->PN[0]=NULL;
48 cmdbuf->PS[0]=NULL;
49 } /* getprcmd() */
50 getlline(fp, lbuf) /* 从定义文件中取一定行 */
51 FILE *fp;
52 char *lbuf;
53 {
54     register i=0;
55     char tamps[80];
56     if (fgets(lbuf, SIZELINE, fp)==NULL)
57         return NOTFOUND;
58     i=strlen(lbuf);
59     while (lbuf[i-2]!='\\'){
60         lbuf[i-2]=NULL;
61         if (fgets(tamps, SIZELINE, fp)==NULL)
62             return NOTFOUND;
63         i=strlen(strcat(lbuf, tamps));
64     }
65     return i;
66 } /* getlline() */

```

May 3 11:07 1993 Page 2

```

67 getnextname(lbuf, nbuf, i) /* 从定义行中取下一终端名 */
68 char *lbuf, *nbuf;
69 {
70     register j=0;
71     while ((nbuf[j++] = lbuf[i++]) != '\n'){
72         if (nbuf[j-1] != '!'){
73             nbuf[j-1] = NULL;
74             return i;
75         }
76         else if (nbuf[j-1] == ':'){
77             nbuf[j-1] = NULL;
78             return 0; /* 最后一个终端名 */
79         }
80     }
81     return NOTFOUND; /* 取不到终端名 */
82 } /* getnextname */
83 getlrecord(lbuf, rbuf, i) /* 从定义行中取下一记录 */
84 char *lbuf, *rbuf;
85 {
86     register j=0;
87     while ((rbuf[j] = lbuf[i++]) != ':'){
88         if (rbuf[j] == '\n') return NOTFOUND; /* 行结束, 取不到记录 */
89         else if (rbuf[j] != ' ') j++;
90         rbuf[j] = NULL;
91         return i;
92     } /* getlrecord() */
93 matchtermname(tname, lbuf) /* 检查终端名与命令定义行是否匹配 */
94 char *tname, *lbuf;
95 {

```

May 3 11:07 1993 Page 3

```

96     register i=0;
97     char xname[SIZETERM];
98     while ((i=getnextname(lbuf, xname, i))!=NOTFOUND)
99         if (strcmp(tname, xname)==0) return 1; /* 终端名匹配 */
100     else if (i==0) return 0; /* 终端名不匹配 */
101     return 0; /* 终端名不匹配 */
102 } /* matchtermname() */
103 getPNandPS(cmdbuf, lbuf) /* 从命令行中取透明打印命令 */
104 struct pcommand *cmdbuf;
105 char *lbuf;
106 {
107     register i=0;
108     char rbuf[SIZECMD];
109     for (; (lbuf[i] != ':') && (lbuf[i] != '\n'); i++);
110     if (lbuf[i] == '\n') goto nullvalue; /* 文件格式不对 */
111     else i++;
112     cmdbuf->PN[0] = '*';
113     cmdbuf->PS[0] = '*';
114     while ((i=getlrecord(lbuf, rbuf, i))!=NOTFOUND){
115         if (strcmp("PN=", rbuf, 3)==0){
116             getcmd(cmdbuf->PN, rbuf);
117             if (cmdbuf->PS[0] != '*') return;
118         }
119         else if (strcmp("PS=", rbuf, 3)==0){
120             getcmd(cmdbuf->PS, rbuf);
121             if (cmdbuf->PN[0] != '*') return;
122         }
123     }
124     nullvalue:
125     cmdbuf->PN[0]=NULL;
126     cmdbuf->PS[0]=NULL;
127 } /* getPNandPS() */
128 getcmd(cmdbuf, rdbuf) /* 从记录中取命令序列 */
129 char *cmdbuf, *rdbuf;
130 {
131     register i=0, j=3;
132     while ((cmdbuf[i++] = rdbuf[j++]) != NULL)
133         if ((cmdbuf[i-1] != 'E') && (i>1) && (cmdbuf[i-2] == '\\')){
134             i--;
135             cmdbuf[i++] = ESC;
136         }
137 } /* getcmd() */

```