## Subject: VT100 V2.3 Possible Bug and Possible Fix
Posted by star on Wed, 03 Dec 1986 06:02:53 GMT
View Forum Message <> Reply to Message

Article-I.D.: vax4.3711
Posted: Wed Dec  3 01:02:53 1986
Date-Received: Sun, 7-Dec-86 04:27:31 EST
Organization: John Fluke Mfg. Co., Inc., Everett, WA
Lines: 25
Keywords: Wecker VT100

I was always perplexed as to why the VT100 program's arrow keys (at least to me), did not work properly when using 'vi' on our vax.  I noted that in the module 'window.c' around line 300-340 that a variable 'keyapp' was used to select an appropriate action.  This was true with the arrow keys also.  I was not quite sure what function 'keyapp' played with the arrow keys but took the liberty to 'rem' out the following code.

```
 case 0x4f: sendchar(27);
      /* if (keyapp) sendchar('O');
      else */ send(']');
      sendchar(code - 11);
      break;
```

This did the trick and the arrow keys now work well with a 'vt100' termcap on the vax and also with 'vi'.

Can someone enlighten me if this is really a fix or did I buger something else up.

--
Dave Whitlock
        {decvax!microsof,uw-beaver,ssc-vax,allegra,lbl-csam}!fluke!star
--John Fluke Mfg. Co., 33031 Schoolcraft Road, Livonia, MI 48150

## Subject: Re: VT100 V2.3 Possible Bug and Possible Fix
Posted by papa on Tue, 09 Dec 1986 03:19:06 GMT
View Forum Message <> Reply to Message

Article-I.D.: bacall.2213
Posted: Mon Dec  8 22:19:06 1986

Date-Received: Wed, 10-Dec-86 05:31:48 EST
References:
Organization: CS&CE Depts, U.S.C., Los Angeles, CA
Lines: 44

```
 >  I was always perplexed as to why the VT100 program's arrow keys (at least
 >  to me), did not work properly when using 'vi' on our vax.  I noted that in
 >  the module 'window.c' around line 300-340 that a variable 'keyapp' was
 >  used to select an appropriate action.  This was true with the arrow keys
 >  also.  I was not quite sure what function 'keyapp' played with the arrow
 >  keys but took the liberty to 'rem' out the following code.
 >
 >
 >   case 0x4f: sendchar(27);
 >        /* if (keyapp) sendchar('O');
 >        else */ send(']');
 >        sendchar(code - 11);
 >        break;
 >
 >  This did the trick and the arrow keys now work well with a 'vt100' termcap
 >  on the vax and also with 'vi'.
 >
 >  Can someone enlighten me if this is really a fix or did I buger something
 >  else up.
 >
 >
 >  --
 >  Dave Whitlock
 >          {decvax!microsof,uw-beaver,ssc-vax,allegra,lbl-csam}!fluke!star
 >  --John Fluke Mfg. Co., 33031 Schoolcraft Road, Livonia, MI 48150
```

The above code seems to be buggy in ANY case. The variable "keyapp" is connected with the fact that one is using the "application" mode instead of the numeric mode.  Unfortunately this is related to the keypad keys and NOT the cursor keys.  The cursor keys have their own SET or RESET state totally unrelated to the state of the keypad.

I have seen the problem with vi and the arrow keys on a "REAL" vt100. The problem is more acute with vi under Berkeley 4.2, while does not exist on vi under ATT System V.  I believe it to be a bug in 4.2 vi.  The EDT editor under VMS (a DEC product) performs properly with the documented escape sequences.

If the above code uses "keyapp" to mean cursor key mode, it would work properly, but in that case it is really using the wrong variable name.

-- Marco Papa
   Felsina Software

(author of the VT100 emulator included in A-Talk 1.1)

---

Subject: Re: VT100 V2.3 Possible Bug and Possible Fix
Posted by wagner on Tue, 09 Dec 1986 15:21:47 GMT
View Forum Message <> Reply to Message

Article-I.D.: utcs.1986Dec9.102147.14725
Posted: Tue Dec 9 10:21:47 1986
Date-Received: Tue, 9-Dec-86 10:32:07 EST
References:
Reply-To: wagner@utcs.UUCP (Michael Wagner)
Organization: University of Toronto Computing Services, general purpose UNIX
Lines: 29
Keywords: Wecker VT100
Checksum: 31425

David Whitlock writes that the arrow keys don't seem to do the right things
in the VT100 program. I reported something similar a while ago, as a bug
report. I guess Dave Wecker didn't see it. I got back a bunch of mail from
other people, and the net result is...

Dave's program is wrong. However, it only shows for some programs. Moreover,
it isn't clear what is right.

There are really two modes on a VT100. There is keypad application mode, and
another one (whose name escapes me at present). Keypad application mode is
the mode that makes the keypad on the right present different codes than
similarly marked keys in the body of the keyboard. If your application runs
in keyboard application mode, Dave Wecker's VT100 incorrectly switches the
cursor key sequences. The fix that Dave Whitlock gave will correct this
(if that's all your application does). This makes the VT100 program capable
of talking to Series/1s running Yale/Ascii, for instance.

However, this other mode (whose name escaped) is supposed to effect the
cursor keys. I suspect that IBM 7171s use this mode. I hardcoded and
recompiled with both sequences, and neither would make the 7171 happy.
I wish someone would pull out a VT100 manual and set us all straight here.
It seems we're still confused.

Disclaimer: I'm still running VT100 V2.2. Dave Whitlock's comments make me
think that the problem is still not solved in 2.3. Sorry to have
wasted everyone's time if the problem is indeed solved.

Michael Wagner (wagner@utcs)

Subject: Re: VT100 V2.3 Possible Bug and Possible Fix
Posted by Anonymous on Wed, 10 Dec 1986 07:02:28 GMT
View Forum Message <> Reply to Message

Originally posted by: johnth&#64batcomputer.UUCP

Article-I.D.: batcompu.1779
Posted: Wed Dec 10 02:02:28 1986
Date-Received: Sun, 14-Dec-86 02:00:43 EST
References:
Reply-To: johnth@batcomputer.UUCP (john thurtell)
Organization: Theory Center, Cornell University, Ithaca NY
Lines: 24
Keywords: Wecker VT100

In article wagner@utcs.UUCP (Michael Wagner) writes:
>
> There are really two modes on a VT100. There is keypad application mode, and
> another one (whose name escapes me at present). Keypad application mode is
> the mode that makes the keypad on the right present different codes than
> similarly marked keys in the body of the keyboard. If your application runs
> in keyboard application mode, Dave Wecker's VT100 incorrectly switches the
> cursor key sequences.
>
> Michael Wagner (wagner@utcs)

I don't know about Dave's program but a real VT100 does have two modes for
both the keypad and the cursor keys.
The application mode for the keypad is set by ESC=
and numeric mode is set by ESC>
For the cursor keys application mode is set by ESC[?1h
and cursor mode is set by ESC[?1l
There is also VT52 mode which can be turned on by the seq. ESC[?2l
If you don't cover them all (and there may be more) you will
of course find applications which don't work.

John Thurtell johnth@batcomputer.UUCP UUCP
thurtell@cheme.tn.cornell.edu ARPA
p8tj@cornelld.BITNET BITNET

Subject: Re: VT100 V2.3 Possible Bug and Possible Fix
Posted by wagner on Wed, 10 Dec 1986 15:18:05 GMT
View Forum Message <> Reply to Message

Article-I.D.: utcs.1986Dec10.101805.18242
Posted: Wed Dec 10 10:18:05 1986
Date-Received: Wed, 10-Dec-86 11:28:00 EST
References:
Reply-To: wagner@utcs.UUCP (Michael Wagner)
Organization: University of Toronto Computing Services, general purpose UNIX
Lines: 7
Keywords: Wecker VT100
Checksum: 22206

In my previous posting, I said that the vt100 had two modes.  That's not really what I meant to say (foolish me).  I meant to say that the vt100 had two different mode switches (for a total of 4 modes, I suppose...I don't know if it is valid to have both alternate modes on at the same time).

Michael

---

Subject: Re: VT100 V2.3 Possible Bug and Possible Fix
Posted by chas on Thu, 11 Dec 1986 16:11:28 GMT
View Forum Message <> Reply to Message

Article-I.D.: gtss.162
Posted: Thu Dec 11 11:11:28 1986
Date-Received: Sat, 13-Dec-86 22:54:08 EST
References:
Reply-To: chas@gtss.UUCP (Charles Cleveland)
Organization: Georgia Tech Surface Studies
Lines: 250

In article  wagner@utcs.UUCP (Michael Wagner) writes:

> 
> There are really two modes on a VT100.  There is keypad application mode, and
> another one (whose name escapes me at present).  Keypad application mode is
.
.
.
> However, this other mode (whose name escaped) is supposed to effect the
> cursor keys.  I suspect that IBM 7171s use this mode.  I hardcoded and

> recompiled with both sequences, and neither would make the 7171 happy.
> I wish someone would pull out a VT100 manual and set us all straight here.
> It seems we're still confused.

mjp@spice.cs.cmu.edu (Michael Portuesi) did in October, though I didn't
pay any attention to his article until today when I found it covered in
dust in a corner of one of my directories.

I did.  I just posted diffs.  Maybe they didn't make it out alive.  It's
fairly short so I'll repost it here.  Would someone (decide who among
yourselves) let me know if this gets out?  My news/mail feed and I have been
having some problems lately which is why I'm worried about the last posting.

I would especially be interested to know if this works or not on the 7171
or any other system where someone using a genuine vt100 can verify that it
ought to work.  I just did what the manual said.  It works for me on machines
that want normal cursor keys and ones that want application cursor keys,
and which set the terminal to the mode of their choice.  Someday I may add
a menu item to toggle keypad app. mode and one for cursor app. mode so I
can set them myself -- but this is trivial.

These are context diffs.  It you don't know what that is, just look at it
and you will probably find it easy to interpret.  There's much less to
change that the number of lines remaining in this posting might lead you
to think -- most of them are context.

```
----------------------------------------
diff -c remote.c remote.c.fixed
*** remote.c  Mon Nov 24 19:00:56 1986
--- remote.c.fixed Mon Dec  8 21:15:59 1986
***************
*** 88,94

        case 'c':    /* Reset */
   top = MINY; bot = MAXY; savx = MINX; savy = MINY;
!  curmode = FS_NORMAL; keyapp = FALSE;
   inesc = -1;
   a[0] = 0; a[1] = 0; sa[0] = 0; sa[1] = 0;
   emit(12);

--- 88,94 -----

        case 'c':    /* Reset */
   top = MINY; bot = MAXY; savx = MINX; savy = MINY;
!  curmode = FS_NORMAL; keyapp = FALSE; curapp = FALSE;
   inesc = -1;
   a[0] = 0; a[1] = 0; sa[0] = 0; sa[1] = 0;
   emit(12);
```

```
***************
*** 267,273

        case 'h':    /* Set parameter */
  if (private == 0 && p[0] == 20)  nlmode = 1;
! else if (private == '?' && p[0] == 7) p_wrap = 1;
  return;

        case 'l':    /* Reset parameter */

--- 267,276 -----

        case 'h':    /* Set parameter */
  if (private == 0 && p[0] == 20)  nlmode = 1;
! else if (private == '?'){
!  if( p[0] == 7) p_wrap = 1;
!  else if( p[0] == 1) curapp = 1;
! }
  return;

        case 'l':    /* Reset parameter */
***************
*** 272,278

        case 'l':    /* Reset parameter */
  if (private == 0 && p[0] == 20)  nlmode = 0;
! else if (private == '?' && p[0] == 7) p_wrap = 0;
  return;

        case 'x':

--- 275,284 -----

        case 'l':    /* Reset parameter */
  if (private == 0 && p[0] == 20)  nlmode = 0;
! else if (private == '?'){
!  if(p[0] == 7) p_wrap = 0;
!  else if( p[0] == 1) curapp = 0;
! }
  return;

        case 'x':
diff -c vt100.c vt100.c.fixed
*** vt100.c  Mon Nov 24 19:00:24 1986
--- vt100.c.fixed Mon Dec  8 21:12:01 1986
***************
*** 61,66
      y      =     MINY;
      curmode   =     FS_NORMAL;
      keyapp    =     0;
      script_on =     FALSE;
      script_wait=    TRUE;
      SetAPen(mywindow->RPort,1L);

--- 61,67 -----
      y       =     MINY;
      curmode   =     FS_NORMAL;
      keyapp    =     0;
+     curapp    =     0;
      script_on =     FALSE;
      script_wait=    TRUE;
      SetAPen(mywindow->RPort,1L);
diff -c vt100.h vt100.h.fixed
*** vt100.h  Mon Nov 24 19:00:19 1986
--- vt100.h.fixed Mon Dec  8 21:23:37 1986
***************
*** 159,165
  UBYTE  *BeepWave;
  UBYTE  Audio_AllocMap[4] = { 1, 8, 2, 4 };
  int want_message;
! int x,y,curmode,keyapp;
  int MINX = 0;
  int MAXX = 632;
  int MINY = 14;

--- 159,165 -----
  UBYTE  *BeepWave;
  UBYTE  Audio_AllocMap[4] = { 1, 8, 2, 4 };
  int want_message;
! int x,y,curmode,keyapp,curapp;
  int MINX = 0;
  int MAXX = 632;
  int MINY = 14;
***************
*** 254,260
  extern char *rs_in;
  extern struct IOExtSer *Write_Request;
  extern char rs_out[2];
! extern int x,y,curmode,keyapp;
  extern int MINX,MAXX,MINY,MAXY,top,bot,savx,savy;
  extern int savmode,nlmode,alt,savalt,a[2],sa[2];
  extern int inesc,inctrl,private,badseq,maxcol;

--- 254,260 -----
  extern char *rs_in;
  extern struct IOExtSer *Write_Request;
  extern char rs_out[2];
! extern int x,y,curmode,keyapp,curapp;
  extern int MINX,MAXX,MINY,MAXY,top,bot,savx,savy;
  extern int savmode,nlmode,alt,savalt,a[2],sa[2];
  extern int inesc,inctrl,private,badseq,maxcol;
diff -c window.c window.c.fixed
*** window.c  Mon Nov 24 19:01:00 1986
--- window.c.fixed Mon Dec  8 21:07:41 1986
***************
*** 310,316
   case 0x4d:
   case 0x4e:
   case 0x4f: sendchar(27);          /* cursor keys */
!     if (keyapp) sendchar('O');
      else sendchar('[');
      sendchar(code - 11);
      break;

--- 310,316 -----
   case 0x4d:
   case 0x4e:
   case 0x4f: sendchar(27);          /* cursor keys */
!     if (curapp) sendchar('O');
      else sendchar('[');
      sendchar(code - 11);
      break;
---------------------------------------
end of patches
---------------------------------------
```

```
From gatech!seismo!rochester!pt.cs.cmu.edu!spice.cs.cmu.edu!mjp Thu Oct 23 12:58:34 EDT 1986
Article 1516 of net.micro.amiga:
Relay-Version: version B 2.10.3 4.3bsd-beta 6/6/85; site gtss.UUCP
Path: gtss!gatech!seismo!rochester!pt.cs.cmu.edu!spice.cs.cmu.edu!mjp

Newsgroups: net.micro.amiga
Subject: VT100 cursor keys
Message-ID:
Date: 21 Oct 86 18:48:24 GMT
Date-Received: 23 Oct 86 02:40:37 GMT
Reply-To: mjp@spice.cs.cmu.edu (Michael Portuesi)
Distribution: net
Organization: Carnegie-Mellon University, CS/RI
Lines: 43
Keywords:
```

> Description:
> Cursor keys send wrong sequence when the keypad is in application mode

>
> Suggested Fix:
> in window.c, remove the checking for 'keyapp' in the cursor key code.
> always send the sequence currently in the 'else' logic.

A real VT100 has two distinct application modes, Keypad Application Mode and Cursor Key Mode. Keypad Application mode applies only to the keypad; Cursor Key Mode applies only to the cursor keys. Thus the suggested fix above is correct. One should not tamper with the cursor keys when Keypad Application Mode is selected. From the VT100 User Guide:

| Cursor Key | VT52 | ANSI Mode and Cursor | ANSI Mode and Cursor |
|---|---|---|---|
| (Arrow) | Mode | Key Mode Reset | Key Mode Set |
| Up | ESC A | ESC [ A | ESC O A |
| Down | ESC B | ESC [ B | ESC O B |
| Left | ESC C | ESC [ C | ESC O C |
| Right | ESC D | ESC [ D | ESC O D |

The following control sequences enter/exit Cursor Key Mode and Keypad Application Mode:

| Name | Set | Reset |
|---|---|---|
| Cursor Key Mode | ESC [ ? 1 h | ESC [ ? 1 l |
| Keypad Application Mode | ESC = | ESC > |

> Michael Wagner (wagner@utcs)
--

+------------------------------------------------------------------------+
| Mike Portuesi |
| Carnegie-Mellon University Computer Science Department |
| |
| ARPA: mjp@spice.cs.cmu.edu |
| UUCP: {harvard | seismo | ucbvax | decwrl}!spice.cs.cmu.edu!mjp |
| |
| "Talking about music is like dancing about architecture" |
| --Laurie Anderson, "Home of the Brave" |
+------------------------------------------------------------------------+

--
Charles Cleveland chas@ss.physics.gatech.edu
Georgia Tech School of Physics
Atlanta, GA 30332 Georgia Tech Surface Studies

...!{akgua,allegra,amd,hplabs,ihnp4,masscomp,ut-ngp,rlgvax,sb1,
uf-cgrl,unmvax,ut-sally}!gatech!gtss!chas