Re: How to talk to hardware devices in C

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-08/msg00953.html

- *From*: Flash Gordon <spam@xxxxxxxxxxxxxxxxx>
- *Date*: Thu, 09 Aug 2007 19:33:54 +0100

Bliton wrote, On 09/08/07 18:34:

> On Aug 6, 3:10 am, x01001x <xem...@xxxxxxxxxxxx> wrote:
>
> > When programming in C (not C++) how does one send information to a
> > hardware device such as a video card or modem? How is this done in
> > Linux C programming versus Microsoft C programming?
>
> ^^^^^^

> Take a look at following C code which directly communicate with
> Monitor and Fills the entire screen with letter 'A'. You don't need
> the printf statement.
>
> #include<stdio.h>
> #include<conio.h>

conio.h is traditionally associated with some compilers for DOS and the OP asked about Linux.

> void main()
> {
> int i;
> char far *vidmem=0xB8000000;
> for(i=0;i<=3999; i=i+2)
> *(vidmem+i)='A';
> }

markg@brenda:~$ gcc -ansi -pedantic t.c
t.c:2:18: error: conio.h: No such file or directory
t.c: In function main :
t.c:7: warning: ISO C forbids nested functions
t.c:7: error: expected = , , , ; , asm or __attribute__ before * token
t.c:7: error: vidmem undeclared (first use in this function)
t.c:7: error: (Each undeclared identifier is reported only once
t.c:7: error: for each function it appears in.)

```
t.c:5: warning: return type of  main  is not  int
markg@brenda:~$
```

Doesn't seem to work too well on a Linux box which is what the OP asked about. If I remove the inclusion of conie.h (and the unneeded inclusion of stdio.h), which is not needed as well as being non−standard, remove the use of 'far' which is also non−standard, add in the cast required to convert an integer to a pointer and correct the return type of main we get:

```
markg@brenda:~$ cat t.c
int main(void)
{
int i;
char *vidmem=(char*)0xB8000000;
for(i=0;i<=3999; i=i+2)
*(vidmem+i)='A';
return 0;
}
markg@brenda:~$ gcc −ansi −pedantic −Wall −Wextra −O t.c
markg@brenda:~$ ./a.out
Segmentation fault (core dumped)
markg@brenda:~$
```

So it still fails spectacularly. Finally, I happen to remember that not all video cards had the video memory starting at the same location in all modes, so even on DOS it might not always do what you thought.

> for more C programs and assembly language codes which directly
> communicates with hardware go to
>
> http://geocities.com/t1softwares/cprg.htm

The other examples I looked at from this were also examples of how NOT to write C and C++ programs (some files have the extension .CPP) since none of them will compile on this machine.

> and http://geocities.com/t1softwares/asemprg.htm

I can't be bothered to brush off my assembler skills to see if the assembler is as bad.

−−
Flash Gordon
.