

Re: function pointers

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2008-01/msg01150.html

- *From:* "cr88192" <cr88192@xxxxxxxxxxxx>
 - *Date:* Wed, 9 Jan 2008 13:12:34 +1000
-

"jacob navia" <jacob@xxxxxxxxxxxx> wrote in message
[news:fm1452\\$kjv\\$1@xxxxxxxxxxxx](news:fm1452kjv1@xxxxxxxxxxxx)

Ian Collins wrote:

CBFalconer wrote:

Ian Collins wrote:

cr88192 wrote:

.... snip ...

on linux, this option works
better though, since linux
crashes
give a general
debugger-usable core dump
(sadly absent on
windows...).

I pity those poor windows developers :)

Why? I get all that ability. Of course, I am using gcc under
DJGPP on winders.

You prove my point, jumping through hoops to get a useful environment.

I am programming in debian, I have a customer under linux that
wants a server application.

I installed debian (latest).

Re: function pointers

<snip>

There is no IDE, nor any modern development system. I use vi, gdb, and the compiler, exactly like in 1986 when I first started using Unix.

when running on linux, I usually use gedit or kedit.

The different "IDE"s are all completely crap. No go to definition, no browse by function, no nothing. They are barely able to start the compiler.

There is only one debugger: gdb. All other "debuggers" are just software on top of that, but since they did not integrate the debugger source code into the IDE they offer NO extra functionality but they offer a layer of BUGS that is quite incredible.

back when I was young, I was developing using a version of Turbo C. worked pretty, well, and I developed on DOS.

I forget the specifics, TC was really cheap in the early 90s, back when I was young (early elem, technically, but coding was cool, and parents got for me a compiler and some books...).

this was long ago, and I operated pretty solidly in DOS. later on in this part of my life, Win95 was around and gaining popularity, but I mostly used Win3.11, as my computer at the time was a little weak for 95 (I think I had a few other gripes with 95 as well).

in general though, I developed primarily in DOS land...

eventually, I switched over to DJGPP (moving along into the later 90s).

at this point, I went over to Linux, which had remained my primary OS. on linux (back then, my preferred distro was Slackware), the main compiler I used was, of course, GCC.

for a long time, my primary editor was jmacs, and I think I mostly used the thing in console mode (this was before the rise of KDE and GNOME, on X there was FVWM...).

Re: function pointers

Re: function pointers

later on, it was Mandrake, KDE, and GNOME (I forget exactly when these came about).

around 2002 or 2003 or so, when the problems that endlessly plague Linux (lame driver support, ...) eventually frustrated me into a switch (I used Win 2000, as IMO 98 was worse crap than 95, but oddly, my brother liked 98 more, I guess because most games tended to break on 2000...).

I liked an OS that didn't crash endlessly...

and, on Windows, I used GCC (primarily Cygwin).

and, several years later, I switched over primarily to MinGW.

and, as such, I have never really had much use for, or dependence on, IDEs...

however, I have often wished that some things were a little nicer though (I have been engineering some of my "own" solutions though, namely, a slightly "different" way of approaching compilers).

in a very technical sense, I have a kind of C-based VM.
it is torn between the worlds of static compilation and being a VM...

technically the inverse of the JVM, or LLVM, though, which "compile" then "run in the VM", rather, the VM *is* the compiler, and the linker, so it starts pulling in and linking static libraries, objects, and source files, at runtime...

Printer doesn't work, but this is no great pain since I can always reboot and print under windows.

There is no obvious way to write to a DVD to make a backup. I looked at some "HOWTOS" (that goes as documentation this days), but they were outdated. Obviously, for Unix gurus I am just a loser that doesn't know how to use the system, and that is right. I did not want to learn how to write obscure commands to make the sound work, to download software from the internet to make the cd writer work, to study CUPS and make the printer work, to study GRUB and see why my changes to the configuration file are overwritten...

yes.

I remember back when CD recorders were new as well...

Re: function pointers

Re: function pointers

never have done a whole lot of burning in linux though.
even when it works, mkisofs+cdrecord is a hassle to get to work...

Yes, I *could* learn all that if I wanted to... I learned enough of GRUB to find out that the grub system mixed the hard disks and that I had to replace hd1 by hd0 using the built in editor (took me 2 hours) of googling, reading, etc.

That was a FATAL error and I HAD to fix it. But I can live without sound, printer, screen saver, and I can develop using vi+gcc+gdb. It is horrible but if it is well paid I do not give a damm. I have programmed in worst environment than this one.

At least, vi works, and I do not have to use ed

:~)

I am a linux user since it came out with the first slackwares...
AND I will go on using it, mind you. As long as there are customers under that stuff, and as long as I can make it do what I need, it is ok!

well, my comment is that gedit or kedit may be worthwhile.

gedit has tabs and syntax highlighting, and kedit is more like notepad...

I sit and watch in hope that linux's driver support eventually stops being so terrible...
if I use the OS, sound and network are at least a few major things I like to work...

that is probably the major reason I am not using linux right now...

—
jacob navia
jacob at jacob point remcomp point fr
logiciels/informatique
<http://www.cs.virginia.edu/~lcc-win32>

Re: function pointers