Subject: Cross-compilation for MacOS on Linux
Posted by Novo on Sun, 24 Jan 2021 13:22:36 GMT
View Forum Message <> Reply to Message

I'm trying to set up cross-compilation for MacOS on Linux using osxcross.
I'm able to compile code using COMMON_FLAGS = "OSX POSIX BSD".
But linker is still using libraries for Linux.

-lX11
-lXinerama
-lXrender -lX11
-lXft
-lXdmcp
-lfontconfig -lfreetype
-lxcb
-lXext
-lgtk-3 -lgdk-3 -lpangocairo-1.0 -lpango-1.0 -lharfbuzz -latk-1.0 -lcairo-gobject -lcairo
-lgdk_pixbuf-2.0 -lgio-2.0 -lgobject-2.0 -lglib-2.0
-lnotify -lgdk_pixbuf-2.0 -lgio-2.0 -lgobject-2.0 -lglib-2.0
-lexpat
-lpng16 -lz
-lz -lpthread

Could you fix that, please?

TIA

Subject: Re: Cross-compilation for MacOS on Linux
Posted by mirek on Thu, 04 Feb 2021 19:41:59 GMT
View Forum Message <> Reply to Message

Are you using umk or makefile?

Subject: Re: Cross-compilation for MacOS on Linux
Posted by Novo on Thu, 04 Feb 2021 20:29:26 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 04 February 2021 14:41Are you using umk or makefile?
umk.
I believe this can be done similar to porting clang-mingw to Linux.
I had same problem back then.

Subject: Re: Cross-compilation for MacOS on Linux
Posted by mirek on Sun, 07 Feb 2021 17:31:38 GMT
View Forum Message <> Reply to Message

Hopefully fixed.

Subject: Re: Cross-compilation for MacOS on Linux
Posted by Novo on Mon, 08 Feb 2021 18:29:33 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 07 February 2021 12:31Hopefully fixed.
Thanks a lot!
Interestingly, this change broke compilation. Depending on passed common flags I get different error messages.
For "OSX POSIX BSD" I get
Error executing /home/ssg/dvlp/cpp/code/lang/osxcross/target/bin/o64-clang++ -c -I/usr/include/freetype2 -I/usr/include/libpng16
I believe POSIX and BSD are not supposed to be used in this case.
And for the plain OSX I get
osxcross: warning: cannot find clang intrinsic headers; please report this issue to the OSXCross project
I didn't get this error message before ...
The problem was introduced in a commit uppsrc: MacOS 11.2 fixes
The change:

```
Vector<String> SplitDirs(const char *s) {
#ifdef PLATFORM_POSIX
 return Split(s, [](int c) { return findarg(c, ';', ':') >= 0 ? c : 0; });
#else
 return Split(s, ';');
#endif
}
```

This is weird.
If I restore previous
return Split(s, ';');

I get a linker problem:

```
Undefined symbols for architecture x86_64:
  "Upp::PdfDraw__initializer()", referenced from:
      Upp::PdfDraw__initialize_struct::PdfDraw__initialize_struct() in CtrlLib$blitz.o
  "Upp::PdfDraw::DrawLineOp(int, int, int, int, int, Upp::Color)", referenced from:
      vtable for Upp::PrinterDraw in CtrlLib$blitz.o
  "Upp::PdfDraw::DrawRectOp(int, int, int, int, Upp::Color)", referenced from:
      vtable for Upp::PrinterDraw in CtrlLib$blitz.o
  "Upp::PdfDraw::DrawTextOp(int, int, int, unsigned short const*, Upp::Font, Upp::Color, int, int const*)", referenced from:
      vtable for Upp::PrinterDraw in CtrlLib$blitz.o
```

...

Compilation is fine in this case ...