

Embedded Linux course - Toolchains

Total des points 2/6 ?

What's the difference between ABI and API compatibility ? *

ABI: Application binairies interface. API: Application programming interface

Commentaire

*ABI is about binary compatibility (typically between object code files)
API is about source level compatibility*

✓ How to compile the C library? *

1/1

- ☐ Get a toolchain, then compile your C library
- ☒ That's done during the generation of the toolchain



Commentaire

Correct. The C library is always compiled by the script generating the toolchain and included in the toolchain (except if you have a baremetal toolchain). That's why you don't need to compile the C library once you have the toolchain.



✗ Can you execute a static binary in a root filesystem compiled with a different toolchain ? *

0/1

☐ Yes

☒ No

✗

Bonne réponse

☒ Yes

Commentaire

Wrong. A static binary is self contained, so it will work in any root filesystem, as long as the kernel supports this binary format.

✗ Can you always compile a recent kernel with a toolchain with older kernel headers ? *

0/1

☒ Yes, always

✗

☐ Yes, most of the time

☐ No, most of the time

☐ Never

Bonne réponse

☒ Yes, most of the time

Commentaire

The kernel being a standalone executable, it doesn't use the kernel headers in the toolchain. So, any gcc compiler should do.

The case where this could fail is when the compiler is too old. You cannot compile the Linux kernel with too old gcc versions (see <https://www.kernel.org/doc/html/latest/process/changes.html>).



✗ Can you run an executable build by a toolchain with recent kernel headers, on a system with an older kernel ? *

0/1

- ☐ Yes, always
- ☐ Yes, most of the time
- ☒ No, most of the time
- ☐ Never

✗

Bonne réponse

- ☒ Yes, most of the time

Commentaire

As long as the executable doesn't use recent system calls (not supported in the older kernel), it will be able to run fine. This applies to most normal C programs not using recent system calls.

✓ A Windows x86 toolchain generating code for ARM Linux, and created on 1/1 x86 Linux, is: *

- ☐ A native toolchain
- ☐ A cross toochain
- ☐ A cross-native toolchain
- ☒ A Canadian cross toolchain

✓

Commentaire

This is considered as a Canadian cross toolchain as 3 types of platforms are involved (considering that x86 Linux and x86 Windows are different platforms).



✗ Can an armv7 toolchain generate code for an armv5 target ? *

0/1

☐ Yes

☒ No

✗

Bonne réponse

☒ Yes

Commentaire

Wrong. Even if by default it's optimized for a given ARM CPU, any gcc compiler for ARM can generate optimized code for another target CPU, if given the right options.

Ce formulaire a été créé dans Bootlin.

Google Forms

