



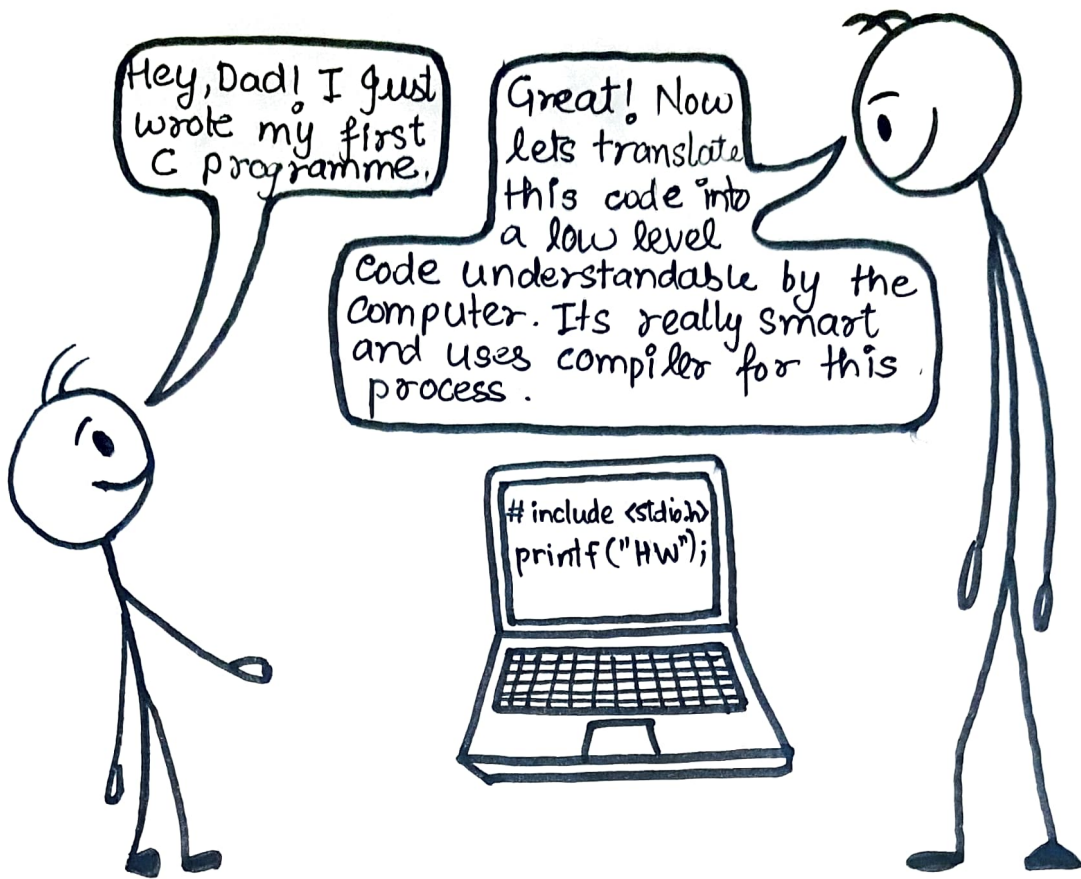
CROSS

COMPTON

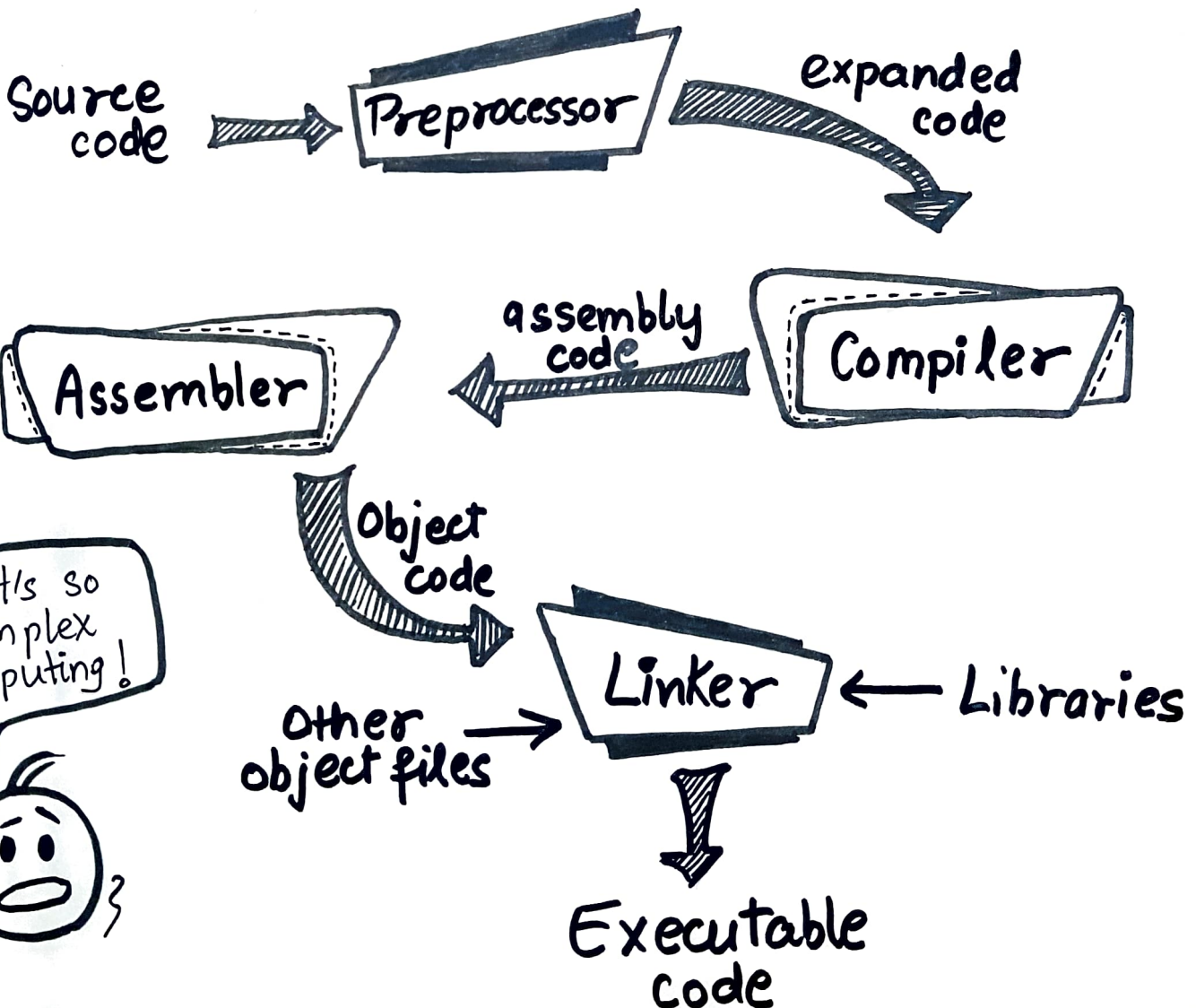
17110160 TANMAEY GUPTA

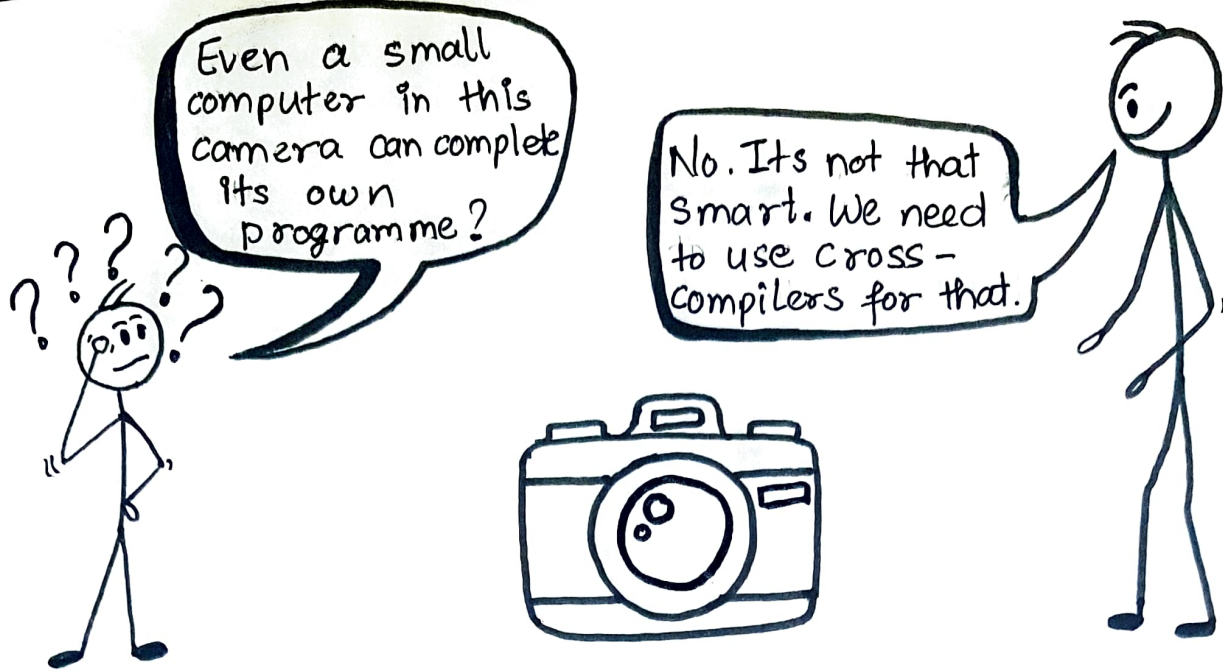
17110104 PATEL URVISHKUMAR





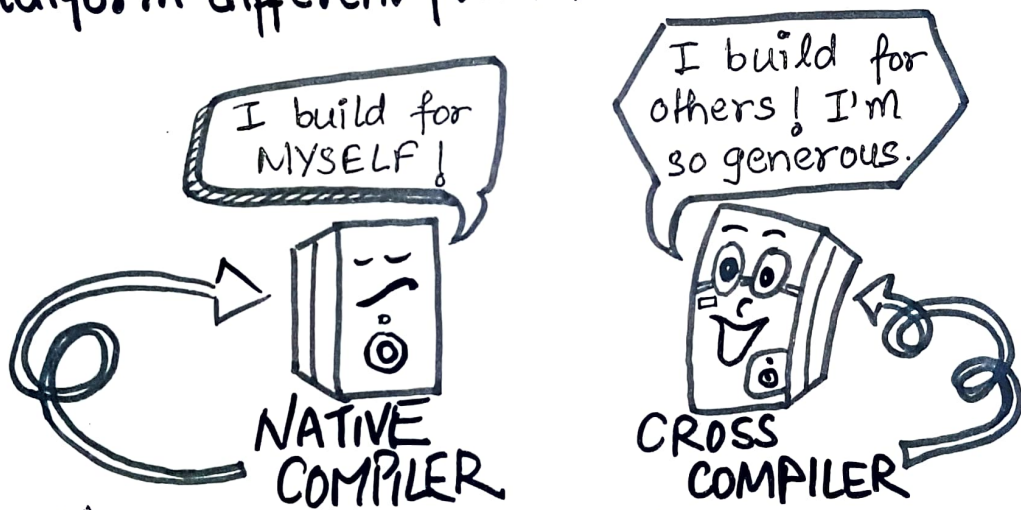
What's a Compiler?





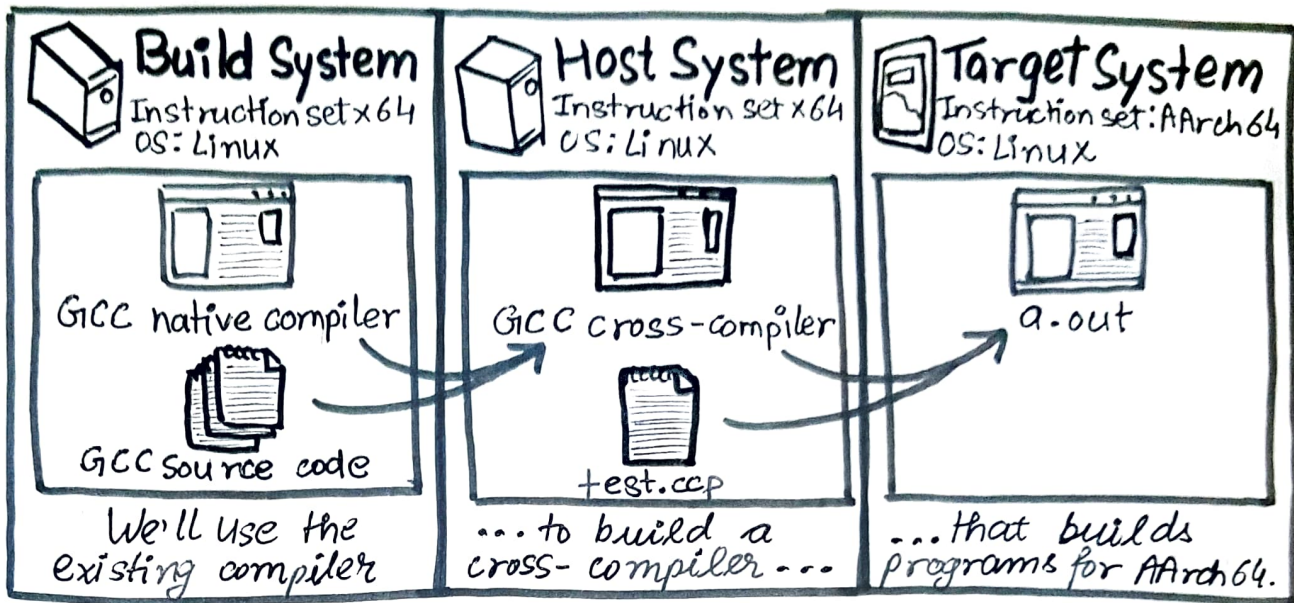
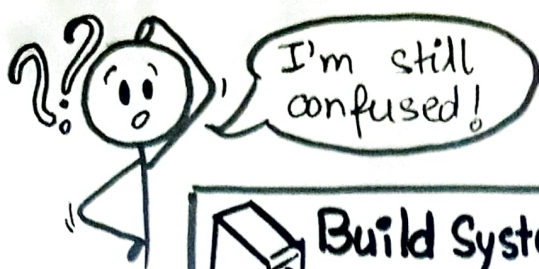
What's a Cross-Compiler?

A compiler used to build source code into binary code that can be executed on a platform different from the one where it was compiled.



APPLICATIONS OF CROSS COMPILER

- Single build environment for different OS platforms in an organisation.
- Bootstrapping and developing other software for a future platform.
- Developing Embedded systems executables since they don't have enough memory and computing resources.
- Simulation of non-native architecture for bench-marking and testing.



I want to build for others too!
How do I start?



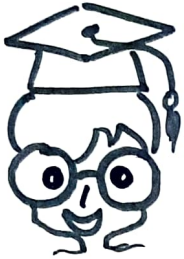
CHECKLIST FOR LINUX CROSS COMPILER TOOL CHAIN

- binutils** It's the collection of binary tools including linker (ld), assembler (as), debugging and analysis tools.
- gcc** GNU compiler collection which provides the main compiler, compiler driver, target libraries and header files for standard libraries.
- Linux Kernel Headers** Definition of system call numbers, various structure types and definitions.
- C library** provides implementation of the POSIX standard functions and other standards and extensions e.g., glibc, musl.

Using GNU autoconf :

```
./configure --build <system A> --host <system B> --target <system C>
```

Generally specifying build and host system is sufficient. If the package being built itself is a cross compiler, target needs to be specified.



Cross compiling a cross-compiler involving build, host and target system is called a Canadian cross. All 3 may have different OS/architecture.

How do I write
<System X>?



CPU architecture
e.g. ARM

Bare-metal
OS (None, Linux, etc.)

<arch>-<vendor>-<os>-<libc/abi>

Ignored by
autoconf (mostly)

Combination of
C library and ABI

E.g. • arm-foo-none-eabi
• mips-img-linux-gnu



Great! Now I
can build executables
for any computer
just from my
computer!