

## O que é uma toolchain

Uma toolchain é um conjunto de pacotes necessários para a criação de software para um dispositivo específico. Uma toolchain pode ser baixada e instalada (utilizando um gerenciador de pacotes ou não) ou podem ser utilizadas com um gerenciador de toolchain.

Ao criar um projeto para uma máquina que utilizará Linux Embarcado, a toolchain faz parte dos primeiros passos (ou ferramentas) necessárias para começar o desenvolvimento deste projeto. Através da utilização da toolchain, pode-se criar: o bootloader; o kernel e o rootfilesystem do linux.

Geralmente, toolchains são baseadas em:

- Projeto GNU e no compilador GCC (gcc, g++, gfortran, etc...);
- Máquinas Virtuais de Baixo Nível e compilador Clang (C, C++, Objective-C e Objective-C++).

O projeto padrão GNU toolchain contém:

- Compiladores capazes de compilar C, C++, Assembly, Java. Produzindo códigos em assembly;
- Binutils que transformam o código assembly em binário, e realizam o link de objetos para criar os arquivos executáveis, dentre algumas funções extras;
- Biblioteca padrão C, implementando APIs do POSIX que são capazes de realizar a comunicação direta com o Kernel;
- Debugger.

## Toolchains nativas e cruzadas

Toolchains podem ser nativas (quando o desenvolvimento e o dispositivo têm sistemas similares) ou cruzados (quando o desenvolvimento e o dispositivo (target) tem sistemas diferentes. Isso tem grande importância já que grande parte dos dispositivos de desenvolvimento em Linux Embarcado são feitos através de compilação cruzada.

Enquanto uma metodologia nativa requer que as atualizações de software sejam restritas e controladas, já que a plataforma de desenvolvimento e o dispositivo de interesse tem que ser sincronizados. O desenvolvimento de uma compilação cruzada requer uma quantidade maior de ferramentas e trabalho.

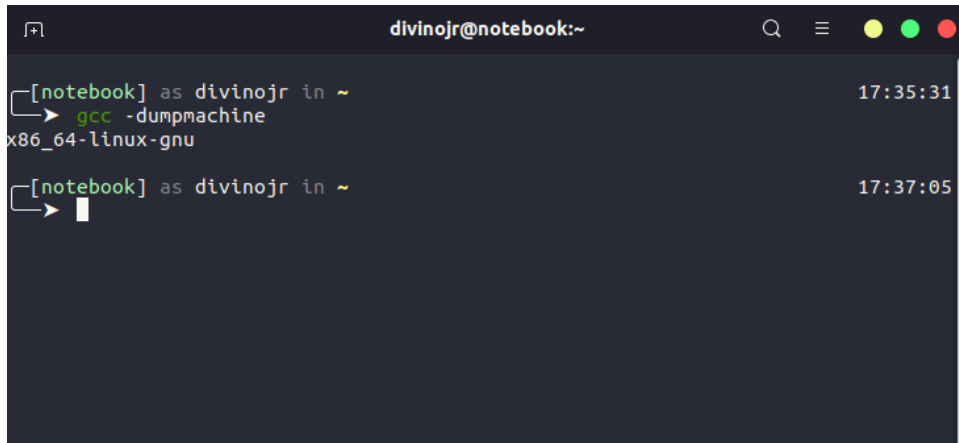
O conhecimento destas ferramentas é importante pois as toolchains devem ser desenvolvidas de forma a comportar os requisitos, como por exemplo:

- CPU: x86\_64, MIPS, ARM, etc...;
- Big/little endian (relacionado com o byte mais significativo e gerenciamento da memória);
- Ponto Flutuante;
- ABI – Interface de Aplicação Binária;

## Identificação das Toolchains utilizando GNU

O projeto GNU utiliza o sistema padrão para identificar as toolchains com o comando abaixo.

`gcc -dumpmachine`



```
divinojr@notebook:~  
[notebook] as divinojr in ~ 17:35:31  
➤ gcc -dumpmachine  
x86_64-linux-gnu  
[notebook] as divinojr in ~ 17:37:05  
➤
```

## ## C library

C libraries contém um conjunto de instruções e funções para chamadas de sistemas para as quais são nomeadas. Onde, o programa principal utiliza essas funções para realizar as chamadas de sistema. Alguns exemplos são:

- glibc -> Uma das melhores implementações da POSIX API;
- eglibc -> Uma ramificação da glibc dedicada para sistemas embarcados (obsoleta e não recebe mais atualizações)
- musl libc -> uma das melhores escolhas para sistemas com pouca memória RAM e/ou armazenamento
- uClibc-ng -> uma ramificação de Clibc desenvolvida para sistemas embarcados e dispositivos móveis que utilizam o microLinux.

## ## Etapas para escolher um toolchain

Ao escolher uma toolchain, usualmente tem-se três opções:

1. Uma toolchain pré-construída;

É uma opção mais simples, porém, menos flexível. Uma escolha recomendada deve incluir, a biblioteca de referência em C desejada e que seja de fácil atualização.

2. Uma toolchain criada do zero antes da instalação;

É uma opção mais complexa, porém, existem vários projetos existentes. Uma abordagem mais simples e menos trabalhosa consiste em usar “crosstool-NG”, que acompanha uma quantidade significativa de scripts úteis.

As ferramentas mencionadas podem ser encontradas acessando:

- Cross Linux From Scratch: <https://trac.ckfs.org>
- Crosstool-NG: <http://crosstool-ng.github.io>
- 3. Uma toolchain criada utilizando uma ferramenta de geração embarcada.

## Instalando o Crosstool-NG

1. Git clone: <https://github.com/crosstool-ng/crosstool-ng.git>

```
divinojr@notebook:~/Desktop/Semana10 17:59:16
[notebook] as divinojr in ~/Desktop
> mkdir Semana10

divinojr@notebook:~/Desktop 17:59:21
[notebook] as divinojr in ~/Desktop
> cd Semana10

divinojr@notebook:~/Desktop/Semana10 17:59:25
[notebook] as divinojr in ~/Desktop/Semana10
> git clone https://github.com/crosstool-ng/crosstool-ng.git
Cloning into 'crosstool-ng'...
remote: Enumerating objects: 41146, done.
remote: Counting objects: 100% (1857/1857), done.
remote: Compressing objects: 100% (1042/1042), done.
remote: Total 41146 (delta 848), reused 1530 (delta 772), pack-reused 39289
Receiving objects: 100% (41146/41146), 15.54 MiB | 13.43 MiB/s, done.
Resolving deltas: 100% (26061/26061), done.

divinojr@notebook:~/Desktop/Semana10 17:59:39
[notebook] as divinojr in ~/Desktop/Semana10
> █
```

2. Mover para o diretório de testes

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng/testing 18:00:20
[notebook] as divinojr in ~/Desktop/Semana10
> ls
crosstool-ng

divinojr@notebook:~/Desktop/Semana10 18:00:20
[notebook] as divinojr in ~/Desktop/Semana10
> cd crosstool-ng

divinojr@notebook:~/Desktop/Semana10/crosstool-ng on (master) ✓
(⌂-v-~) > ls
bash-completion  contrib  docs      licenses.d  packages  scripts
bootstrap        COPYING  issue_template.md  m4         paths.sh.in  testing
config           ct-ng.in kconfig   maintainer  README.md    TODO
configure.ac     debian  LICENSE   Makefile.am  samples

divinojr@notebook:~/Desktop/Semana10/crosstool-ng on (master) ✓
(⌂-v-~) > cd testing

divinojr@notebook:~/Desktop/Semana10/crosstool-ng/testing on (master) ✓
(⌂-v-~) > █
```

3. Mover para o diretório “docker”

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng/testing/doc...  
[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng/testing on (master)✓  
(~) ρ pwd  
/home/divinojr/Desktop/Semana10/crosstool-ng/testing  
[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng/testing on (master)✓  
(~) ρ cd docker  
divinojr in ~/Desktop/Semana10/crosstool-ng/testing/docker on (master)✓  
(~) ρ ls  
alpine3.15  centos-stream9  docker-remove-all.sh  mint20-amd64  
archlinux  common-scripts  fedora35              ubuntu18.04  
centos7    dmgr.sh         gentoo-amd64          ubuntu21.10  
divinojr in ~/Desktop/Semana10/crosstool-ng/testing/docker on (master)✓  
(~) ρ
```

4. Encontrar o diretório da distribuição e mover

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng/testing/doc...  
[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng/testing on (master)✓  
(~) ρ pwd  
/home/divinojr/Desktop/Semana10/crosstool-ng/testing  
[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng/testing on (master)✓  
(~) ρ cd docker  
divinojr in ~/Desktop/Semana10/crosstool-ng/testing/docker on (master)✓  
(~) ρ ls  
alpine3.15  centos-stream9  docker-remove-all.sh  mint20-amd64  
archlinux  common-scripts  fedora35              ubuntu18.04  
centos7    dmgr.sh         gentoo-amd64          ubuntu21.10  
divinojr in ~/Desktop/Semana10/crosstool-ng/testing/docker on (master)✓  
(~) ρ cd ubuntu21.10  
~/Desktop/Semana10/crosstool-ng/testing/docker/ubuntu21.10 on (master)✓  
(~) ρ ls  
Dockerfile  
~/Desktop/Semana10/crosstool-ng/testing/docker/ubuntu21.10 on (master)✓  
(~) ρ
```

5. Abrir o arquivo “Dockerfile” com um editor de texto

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng/testing/doc...
Dockerfile
~/Desktop/Semana10/crosstool-ng/testing/docker/ubuntu21.10 on (master) ✓
(⌂) cat Dockerfile
FROM ubuntu:21.10
ARG CTNG_UID=1000
ARG CTNG_GID=1000
RUN groupadd -g $CTNG_GID ctng
RUN useradd -d /home/ctng -m -g $CTNG_GID -u $CTNG_UID -s /bin/bash ctng

# Non-interactive configuration of tzdata
ENV DEBIAN_FRONTEND noninteractive
ENV DEBCONF_NONINTERACTIVE_SEEN true
RUN { echo 'tzdata tzdata/Areas select Etc'; echo 'tzdata tzdata/Zones/Etc select UTC'; } | debconf-set-selections

RUN apt-get update
RUN apt-get install -y gcc g++ gperf bison flex texinfo help2man make libncurses5-dev \
    python3-dev autoconf automake libtool libtool-bin gawk wget bzip2 xz-utils unzip \
    patch libstdc++6 rsync git
RUN wget -O /sbin/dumb-init https://github.com/Yelp/dumb-init/releases/download/v1.2.5/dumb-init_1.2.5_x86_64
RUN chmod a+x /sbin/dumb-init
```

6. Checar a lista de dependências
7. Acessar o diretório “crosstool-ng”
8. Realizar o git checkout da versão desejada

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (master) ✓
(⌂) git checkout
Your branch is up to date with 'origin/master'.

[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (master) ✓
(⌂) git checkout crosstool-ng-1.24.0
Note: switching to 'crosstool-ng-1.24.0'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at b2151f1d Merge pull request #1182 from stilor/master
```

9. Gerar o bootstrap

```
./bootstrap
[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d) ✓
(๑~๗~๕)~ ./bootstrap
INFO :: *** Generating package version descriptions
INFO :: Master packages: android-ndk autoconf automake avr-libc binutils bison
clog gcc dtc duma elf2flt expat gcc gdb gettext glibc glibc-ports gmp isl libelf lib
biconv libtool linux ltrace m4 make mingw-w64 moxiebox mpc mpfr musl ncurses new
lib strace uClibc zlib
INFO :: Generating 'config/versions/android-ndk.in'
INFO :: Generating 'config/versions/autoconf.in'
INFO :: Generating 'config/versions/automake.in'
INFO :: Generating 'config/versions/avr-libc.in'
INFO :: Generating 'config/versions/binutils.in'
INFO :: Generating 'config/versions/bison.in'
INFO :: Generating 'config/versions/clog.in'
INFO :: Generating 'config/versions/dtc.in'
INFO :: Generating 'config/versions/duma.in'
INFO :: Generating 'config/versions/elf2flt.in'
INFO :: Generating 'config/versions/expat.in'
INFO :: Generating 'config/versions/gcc.in'
INFO :: Generating 'config/versions/gdb.in'
INFO :: Generating 'config/versions/gettext.in'
INFO :: Generating 'config/versions/glibc.in'
INFO :: Generating 'config/versions/glibc-ports.in'
INFO :: Generating 'config/versions/gmp.in'
```

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
INFO :: Generating 'config/versions/mingw-w64.in'
INFO :: Generating 'config/versions/moxiebox.in'
INFO :: Generating 'config/versions/mpc.in'
INFO :: Generating 'config/versions/mpfr.in'
INFO :: Generating 'config/versions/musl.in'
INFO :: Generating 'config/versions/ncurses.in'
INFO :: Generating 'config/versions/newlib.in'
INFO :: Generating 'config/versions/strace.in'
INFO :: Generating 'config/versions/uClibc.in'
INFO :: Generating 'config/versions/zlib.in'
INFO :: *** Generating menu/choice selections
INFO :: Generating arch.in (choice)
INFO :: Generating kernel.in (choice)
INFO :: Generating cc.in (choice)
INFO :: Generating binutils.in (choice)
INFO :: Generating libc.in (choice)
INFO :: Generating debug.in (menu)
INFO :: Generating comp_tools.in (menu)
INFO :: Generating comp_libs.in (menu)
INFO :: *** Gathering the list of data files to install
INFO :: *** Running autoreconf
INFO :: *** Done!

[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d) ✓
(๑~๗~๕)~
```

10. Configurar o enable local

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
[divinojr] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d) ✓
[divinojr] $ ./configure --enable-local
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking how to create a pax tar archive... gnutar
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking whether ln -s works... yes
checking whether install takes --strip-program option... yes
checking for ggrep... no
checking for grep... grep
checking for absolute path to grep... /usr/bin/grep
checking for gegrep... no
checking for egrep... egrep
checking for absolute path to egrep... /usr/bin/egrep
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for sed... /usr/bin/sed
checking for GNU sed... yes
```

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
checking for CFlocaleCopyCurrent... no
checking for GNU gettext in libc... yes
checking whether to use NLS... yes
checking where the gettext function comes from... libc
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for ncursesw via pkg-config... yes
checking for working ncursesw/curses.h... yes
checking for working ncursesw.h... no
checking for working ncurses.h... yes
checking for Curses Panel library with ncursesw/panel.h... yes
checking for Curses Menu library with ncursesw/menu.h... yes
checking for build time... Sun Feb 20 18:30:57 2022
checking if the manual needs to be installed... no
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating paths.sh
config.status: creating kconfig/Makefile
config.status: creating config/configure.in
config.status: creating config.h
config.status: executing depfiles commands

[divinojr] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d) ✓
[divinojr] $
```

## 11. Comando Make

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
config.status: executing depfiles commands

[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d) ✓
(๑~๑~๑) make
/usr/bin/make all-recursive
make[1]: Entering directory '/home/divinojr/Desktop/Semana10/crosstool-ng'
Making all in kconfig
make[2]: Entering directory '/home/divinojr/Desktop/Semana10/crosstool-ng/kconfig'
bison -y -l -b zconf -p zconf -ozconf.c zconf.y
zconf.y:34.1-7: warning: POSIX Yacc does not support %expect [-Wyacc]
  34 | %expect 32
    | ^~~~~~
zconf.y:96.1-11: warning: POSIX Yacc does not support %destructor [-Wyacc]
  96 | %destructor {
    | ^~~~~~
flex -L -Pzconf -ozconf.lex.c zconf.l
/usr/bin/make all-am
make[3]: Entering directory '/home/divinojr/Desktop/Semana10/crosstool-ng/kconfig'
depbase='echo conf.o | sed 's|[^/]*$|.deps/&;s|\.o$||'`; \
gcc -DHAVE_CONFIG_H -I. -I.. -include config.h -DCONFIG_=\"CT_\" -g -O2 -MT c
onf.o -MD -MP -MF $depbase.Tpo -c -o conf.o conf.c && \
mv -f $depbase.Tpo $depbase.Po
depbase='echo zconf.o | sed 's|[^/]*$|.deps/&;s|\.o$||'`; \
```

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng

make[2]: Leaving directory '/home/divinojr/Desktop/Semana10/crosstool-ng/kconfig'
make[2]: Entering directory '/home/divinojr/Desktop/Semana10/crosstool-ng'
( /usr/bin/sed -e 's,[@]docdir[@],/usr/local/share/doc/crosstool-ng,g' -e 's,[@]
pkgdatadir[@],/usr/local/share/crosstool-ng,g' -e 's,[@]pkglibexecdir[@],/usr/lo
cal/libexec/crosstool-ng,g' -e 's,[@]progname[@],'echo ct-ng | sed 's,x,x,'',g'
' | /bin/bash config.status --file=- ) < ct-ng.in >ct-ng-t && chmod a-w,a+x ct-n
g-t && mv -f ct-ng-t ct-ng
/usr/bin/mkdir -p bash-completion && ( /usr/bin/sed -e 's,[@]docdir[@],/usr/loca
l/share/doc/crosstool-ng,g' -e 's,[@]pkgdatadir[@],/usr/local/share/crosstool-ng
,g' -e 's,[@]pkglibexecdir[@],/usr/local/libexec/crosstool-ng,g' -e 's,[@]progna
me[@],'echo ct-ng | sed 's,x,x,'',g' | /bin/bash config.status --file=- ) < ba
sh-completion/ct-ng.in >bash-completion/ct-ng-t && mv -f bash-completion/ct-ng-t
bash-completion/ct-ng
/usr/bin/mkdir -p docs && ( /usr/bin/sed -e 's,[@]docdir[@],/usr/local/share/doc
/crosstool-ng,g' -e 's,[@]pkgdatadir[@],/usr/local/share/crosstool-ng,g' -e 's,[
@]pkglibexecdir[@],/usr/local/libexec/crosstool-ng,g' -e 's,[@]progname[@],'ech
o ct-ng | sed 's,x,x,'',g' | /bin/bash config.status --file=- ) < docs/ct-ng.1.
in >docs/ct-ng.1-t && mv -f docs/ct-ng.1-t docs/ct-ng.1
make[2]: Leaving directory '/home/divinojr/Desktop/Semana10/crosstool-ng'
make[1]: Leaving directory '/home/divinojr/Desktop/Semana10/crosstool-ng'

[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d) ✓
(๑~๑~๑)
```

12. Comando Make Install

13. Iniciar o menu de teste da instalação



```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
[~]
[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d) ✓
(02-05) و ./ct-ng
This is crosstool-NG version 1.24.0

Copyright (C) 2008 Yann E. MORIN <yann.morin.1998@free.fr>
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

See below for a list of available actions, listed by category:

Configuration actions:
  show-config      - Show a brief overview of current configuration
  saveconfig       - Save current config as a preconfigured target
  menuconfig       - Update current config using a menu based program
  nconfig          - Update current config using a menu based program
  oldconfig        - Update current config using a provided .config as base
  upgradeconfig    - Upgrade config file to current crosstool-NG
  extractconfig    - Extract to stdout the configuration items from a
                    build.log file piped to stdin
  savedefconfig    - Save current config as a mini-defconfig to ${DEFCONFIG}
  defconfig        - Update config from a mini-defconfig ${DEFCONFIG}
                    (default: ${DEFCONFIG}=./defconfig)
  show-tuple       - Print the tuple of the currently configured toolchain
```

## Verificando exemplos de toolchains

Após realizar os passos anteriores, podemos verificar uma lista com os exemplos de toolchains utilizando o comando na figura a seguir. Essa lista contém exemplos de configurações que são conhecidos para construir e trabalhar com toolchains.

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
[~]
[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d) ✓
(02-05) و ./ct-ng list-samples
Status Sample name
[L...] aarch64-rpi3-linux-gnu
[L..X] aarch64-unknown-linux-android
[L...] aarch64-unknown-linux-gnu
[L...] aarch64-unknown-linux-uclibc
[L...] alphaev56-unknown-linux-gnu
[L...] alphaev67-unknown-linux-gnu
[L...] arc-arc700-linux-uclibc
[L...] arc-multilib-elf32
[L...] arc-multilib-linux-uclibc
[L...] arm-bare_newlib_cortex_m3_nommu-eabi
[L...] arm-cortex_a15-linux-gnueabi
[L..X] arm-cortexa5-linux-uclicbgnueabi
[L...] arm-cortex_a8-linux-gnueabi
[L..X] arm-cortexa9_neon-linux-gnueabi
[L..X] x86_64-w64-mingw32,arm-cortexa9_neon-linux-gnueabi
[L...] armeb-unknown-eabi
[L...] armeb-unknown-linux-gnueabi
[L...] armeb-unknown-linux-uclicbgnueabi
[L...] arm-multilib-linux-uclicbgnueabi
[L...] arm-nano-eabi
[L...] arm-unknown-eabi
```

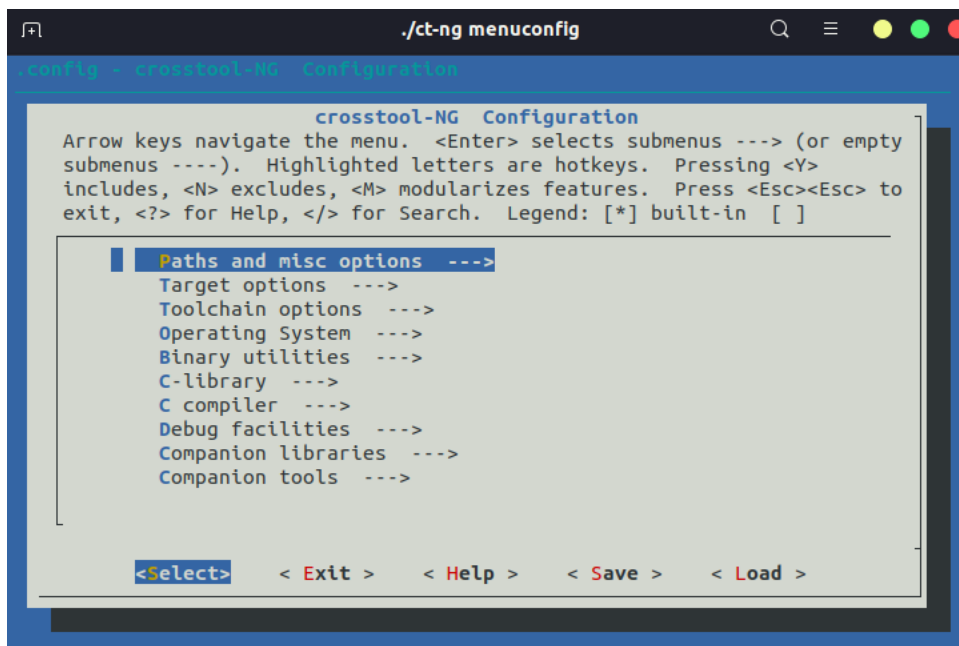
```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
[L...] sparc64-multilib-linux-gnu
[L...] sparc-leon-linux-uclibc
[L...] sparc-unknown-linux-gnu
[L...] x86_64-centos6-linux-gnu
[L...] x86_64-centos7-linux-gnu
[L...] x86_64-multilib-linux-gnu
[L..X] x86_64-multilib-linux-musl
[L...] x86_64-multilib-linux-uclibc
[L..X] x86_64-w64-mingw32,x86_64-pc-linux-gnu
[L...] x86_64-ubuntu12.04-linux-gnu
[L...] x86_64-ubuntu14.04-linux-gnu
[L...] x86_64-ubuntu16.04-linux-gnu
[L...] x86_64-unknown-linux-gnu
[L...] x86_64-unknown-linux-uclibc
[L..X] x86_64-w64-mingw32
[L..X] xtensa-fsf-elf
[L...] xtensa-fsf-linux-uclibc
L (Local)      : sample was found in current directory
G (Global)     : sample was installed with crosstool-NG
X (EXPERIMENTAL): sample may use EXPERIMENTAL features
B (BROKEN)     : sample is currently broken
O (OBSOLETE)   : sample needs to be upgraded

[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d)✓
(๑~๑~๑)و
```

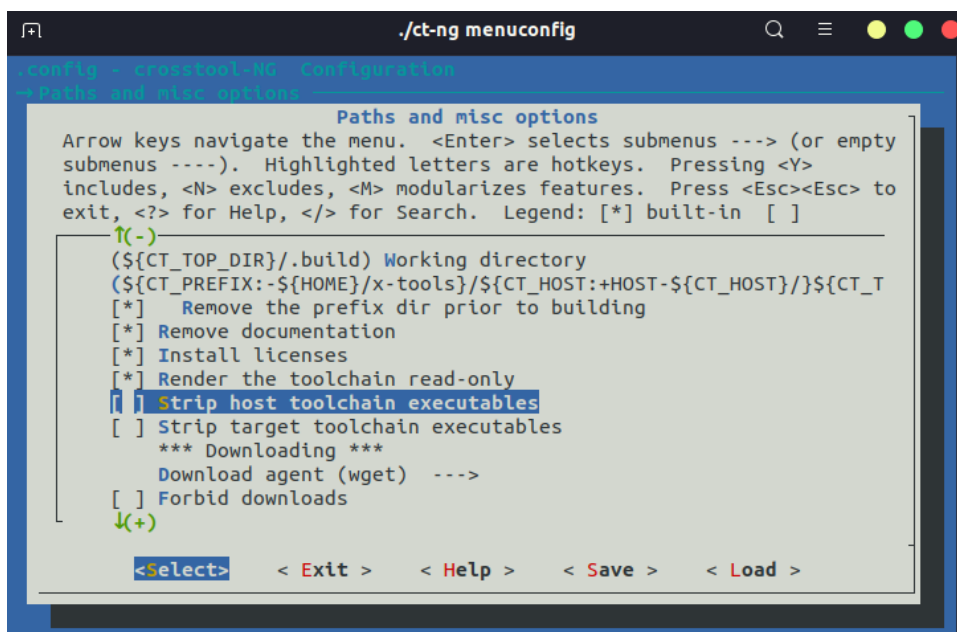
Após escolher um exemplo para o seu dispositivo (se já tiver um).

```
divinojr@notebook:~/Desktop/Semana10/crosstool-ng
[L...] sparc64-multilib-linux-gnu
[L...] sparc-leon-linux-uclibc
[L...] sparc-unknown-linux-gnu
[L...] x86_64-centos6-linux-gnu
[L...] x86_64-centos7-linux-gnu
[L...] x86_64-multilib-linux-gnu
[L..X] x86_64-multilib-linux-musl
[L...] x86_64-multilib-linux-uclibc
[L..X] x86_64-w64-mingw32,x86_64-pc-linux-gnu
[L...] x86_64-ubuntu12.04-linux-gnu
[L...] x86_64-ubuntu14.04-linux-gnu
[L...] x86_64-ubuntu16.04-linux-gnu
[L...] x86_64-unknown-linux-gnu
[L...] x86_64-unknown-linux-uclibc
[L..X] x86_64-w64-mingw32
[L..X] xtensa-fsf-elf
[L...] xtensa-fsf-linux-uclibc
L (Local)      : sample was found in current directory
G (Global)     : sample was installed with crosstool-NG
X (EXPERIMENTAL): sample may use EXPERIMENTAL features
B (BROKEN)     : sample is currently broken
O (OBSOLETE)   : sample needs to be upgraded

[notebook] as divinojr in ~/Desktop/Semana10/crosstool-ng on (b2151f1d)✓
(๑~๑~๑)و
```



Para continuar com a configuração, precisamos remover a flag de “read-only” que nos restringe perante a manipulação do projeto. Fazemos isso acessando “paths and misc options” e desmarcando a opção “Render the toolchain read-only”, conforme apresentado na figura a seguir, utilizando a tecla espaço para desmarcar a opção.



Para iniciar o projeto, utiliza-se o comando a seguir:

```

[notebook] as divinojr in ~/Desktop/Senana10/crostoool-ng on (b2151fid)✓
(⚡~) $ ./ct-ng menuconfig
CONF menuconfig

*** End of the configuration.
*** Execute 'ct-ng build' to start the build or try 'ct-ng help'.

[notebook] as divinojr in ~/Desktop/Senana10/crostoool-ng on (b2151fid)✓
(⚡~) $ ./ct-ng build
[INFO ] Performing some trivial sanity checks
[WARN ] Number of open files 1024 may not be sufficient to build the toolchain; increasing to 2048
[INFO ] Build started 20220220.184637
[INFO ] Building environment variables
[WARN ] Directory '/home/divinojr/src' does not exist.
[WARN ] Will not save downloaded tarballs to local storage.
[EXTRA] Preparing working directories
[EXTRA] Installing user-supplied crostoool-NG configuration
[EXTRA] =====
[EXTRA] Dumping internal crostoool-NG configuration
[EXTRA] Building a toolchain for:
[EXTRA]   build = x86_64-pc-linux-gnu
[EXTRA]   host  = x86_64-pc-linux-gnu
[EXTRA]   target = alphaev4-unknown-elf
[EXTRA] Dumping internal crostoool-NG configuration: done in 0.08s (at 00:01)
[INFO ] =====
[INFO ] Retrieving needed toolchain components' tarballs
[EXTRA] Retrieving 'zlib-1.2.11'
[EXTRA] Verifying SHA512 checksum for 'zlib-1.2.11.tar.xz'
[EXTRA] Retrieving 'gmp-6.1.2'
[EXTRA] Verifying SHA512 checksum for 'gmp-6.1.2.tar.xz'
[EXTRA] Retrieving 'mpfr-4.0.2'
[00:09] /

```

```

[notebook] as divinojr in ~ 18:50:16
→ cd x-tools
[notebook] as divinojr in ~/x-tools 18:50:18
→ ls
alphaev4-unknown-elf
[notebook] as divinojr in ~/x-tools 18:50:19
→ cd alphaev4-unknown-elf
[notebook] as divinojr in ~/x-tools/alphaev4-unknown-elf 18:50:29
→ ls
alphaev4-unknown-elf bin lib
[notebook] as divinojr in ~/x-tools/alphaev4-unknown-elf 18:50:30
→ cd bin
[notebook] as divinojr in ~/x-tools/alphaev4-unknown-elf/bin 18:50:31
→ ls
alphaev4-unknown-elf-ct-ng.config
[notebook] as divinojr in ~/x-tools/alphaev4-unknown-elf/bin 18:50:35
→

```