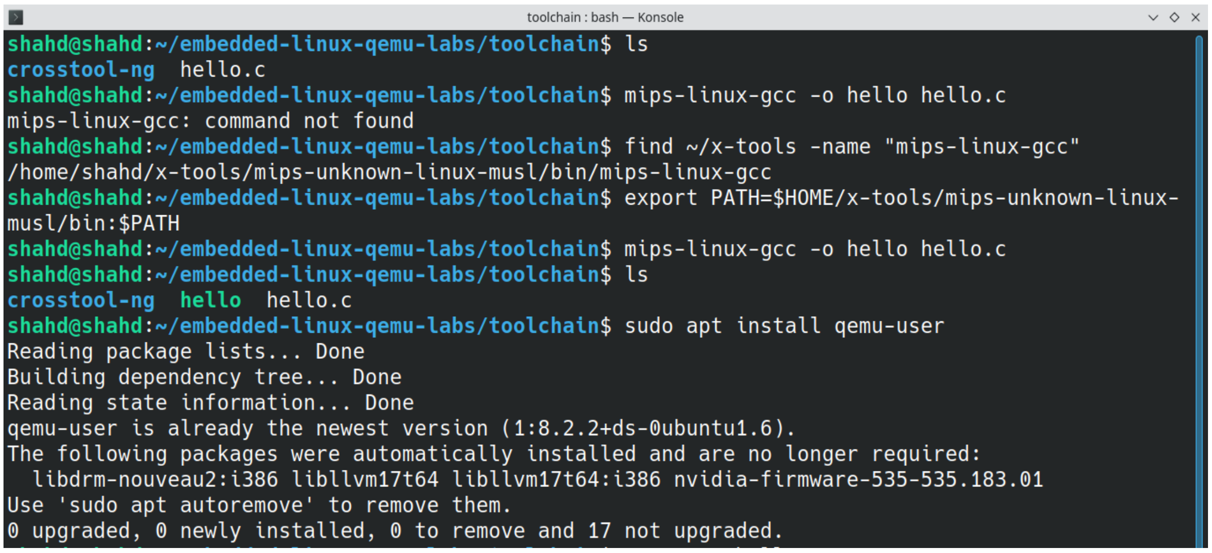
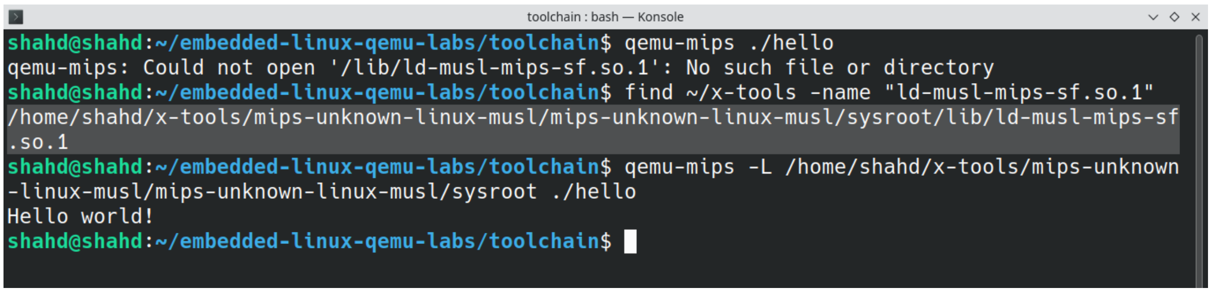
Task_1:

- **Executing the hello.c file, previously downloaded and compiled in the lab, on the MIPS architecture.**



```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ ls
crosstool-ng  hello.c
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ mips-linux-gcc -o hello hello.c
mips-linux-gcc: command not found
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ find ~/x-tools -name "mips-linux-gcc"
/home/shahd/x-tools/mips-unknown-linux-musl/bin/mips-linux-gcc
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ export PATH=$HOME/x-tools/mips-unknown-linux-
musl/bin:$PATH
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ mips-linux-gcc -o hello hello.c
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ ls
crosstool-ng  hello  hello.c
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ sudo apt install qemu-user
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
qemu-user is already the newest version (1:8.2.2+ds-0ubuntu1.6).
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2:i386 libllvm17t64 libllvm17t64:i386 nvidia-firmware-535-535.183.01
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 17 not upgraded.
```



```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ qemu-mips ./hello
qemu-mips: Could not open '/lib/ld-musl-mips-sf.so.1': No such file or directory
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ find ~/x-tools -name "ld-musl-mips-sf.so.1"
/home/shahd/x-tools/mips-unknown-linux-musl/mips-unknown-linux-musl/sysroot/lib/ld-musl-mips-sf
.so.1
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ qemu-mips -L /home/shahd/x-tools/mips-unknown
-linux-musl/mips-unknown-linux-musl/sysroot ./hello
Hello world!
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ 
```

1. Description of the differences between MIPS and ARM architectures and their respective applications.

	MIPS	**ARM**
Design	designed to be very clear and simple in more complex systems. easy to understand and optimize	designed to use little power as much as possible and scale the performance based on task
basic goal	simplicity & efficiency & optimization	low power consumption &high efficiency
commonly used in	severs,older systems	modern systems,phones,small devices,IOT
performance	power-efficient	highly power efficient
license	open	need license

2. Explanation of the difference between cross-compiling toolchain and native tool chain.

- **<u>What is Toolchain?</u>**

  It is a compiler and set of development tools that enables

  the developers to produce a kernel for a specific hardware.

  convert source code into machine code that a computer can run.

- **<u>Consist of:</u>**

  1. Compiler , 2. Assembler, 3. Linker, 4. Libraries,

  5. Debugger.

- **<u>Types of toolchains:</u>**

  1. Cross-compilation toolchain.
  2. Native tool chain.

	Cross-compiling toolchain	**Native tool chain**
What is?	It is toolchain runs on your workstation(host) but it will generate source code for different workstation(target)	It is toolchain runs on your workstation (host)and generate code for your workstation(host)
Use Case	Embedded linux, Mobile devices	General software development
Compiler Examples	arm-cortex9_neon-linux-gnueabihf, mips-unknown-linux-gcc	gcc, clang
Execution	Output runs on a **different** architecture	Output runs on the **same** architecture

Resources:

- Cross Compiling Tool Chains | Sakshi Education
- Difference Between Native Compiler and Cross Compiler - GeeksforGeeks
- A master guide to Linux cross compiling | by Ruvinda Dhambarage | Medium

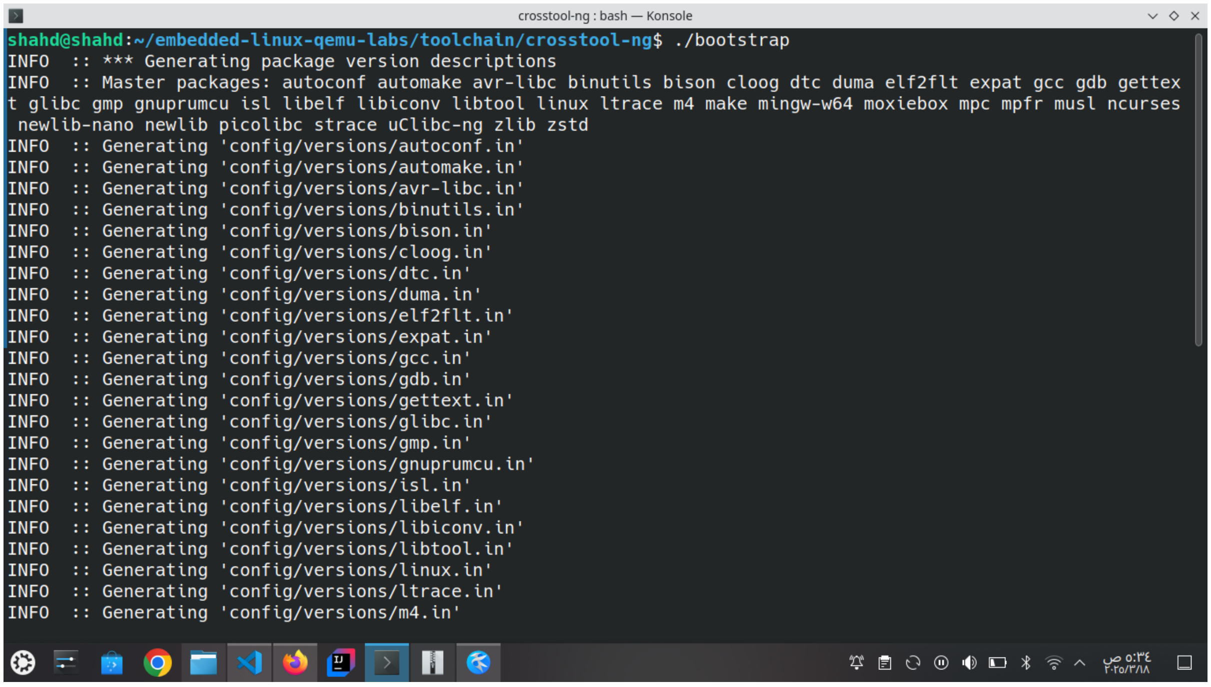
3. Definitions of bootloader, kernel, and filesystem.

bootloader	the first programme that run when open and turn on my device , It loads kernel to memory and checks the hardware components from any issues
kernel	the core of the OS like the brain in the human ,It manages the interaction between software and hardware and allow software to use the hardware resources to make a device that human could interact with
filesystem	system of organizing and storing files in local machine, how they named , who and how could access them, permissions of file based on user. makes dealing and using files easier over all.

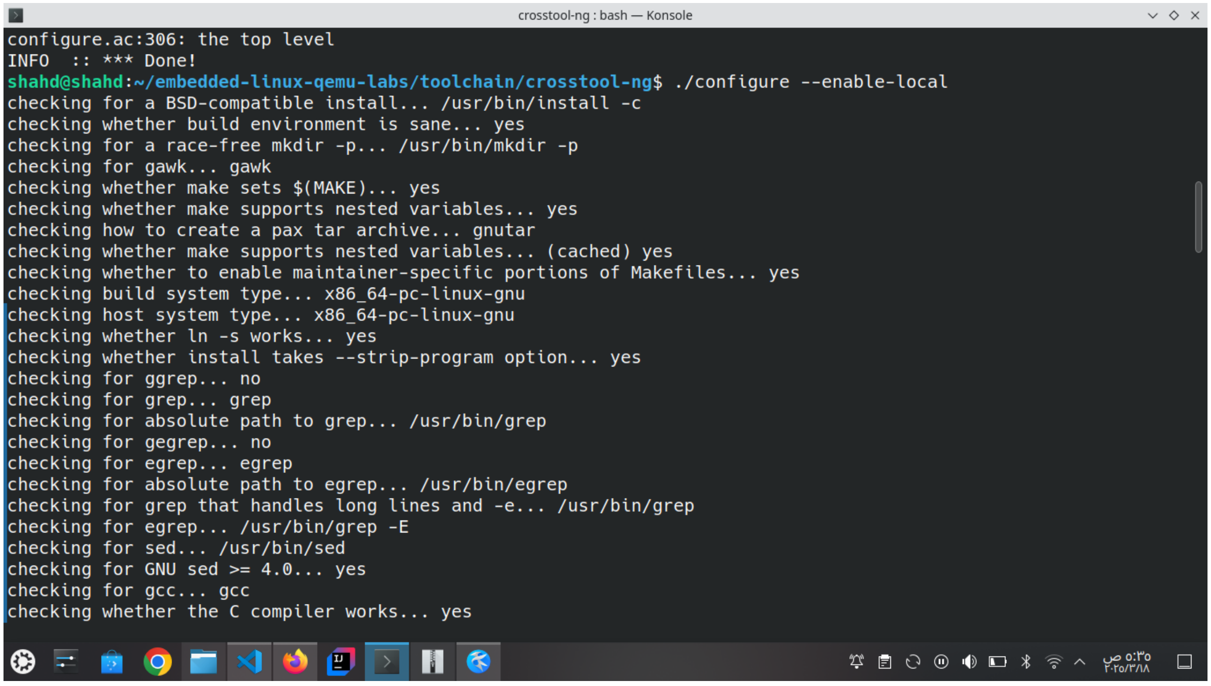
3. Definitions and usage of Kernel modules

- the optional parts of the kernel can be added or removed from it as modules, similar to plugins in programs or applications they add features that couldn’t be done without them but also the programme run successfully without them .They could be added through the run time without needing for restart the computer. They add features that provides more Flexibility ,efficiency and customization to the system.

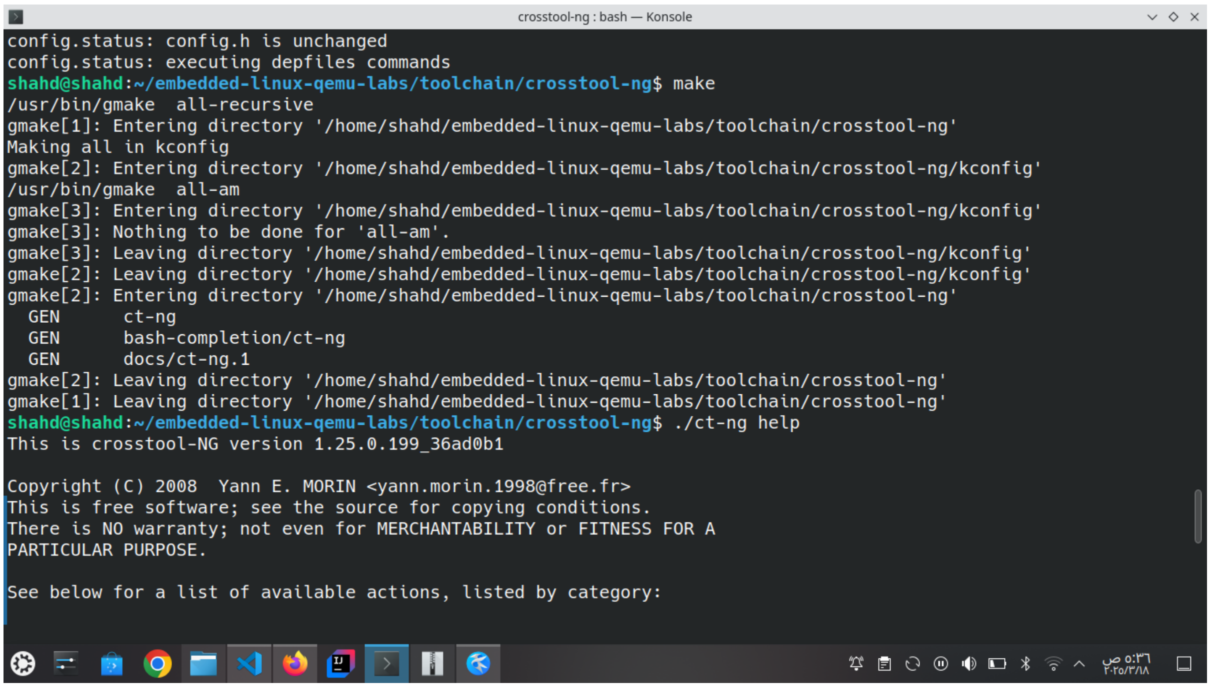
4. Detailed screenshots depicting the steps taken to execute the hello.c code that takes your name as a parameter and displays "hello ${yourName}".



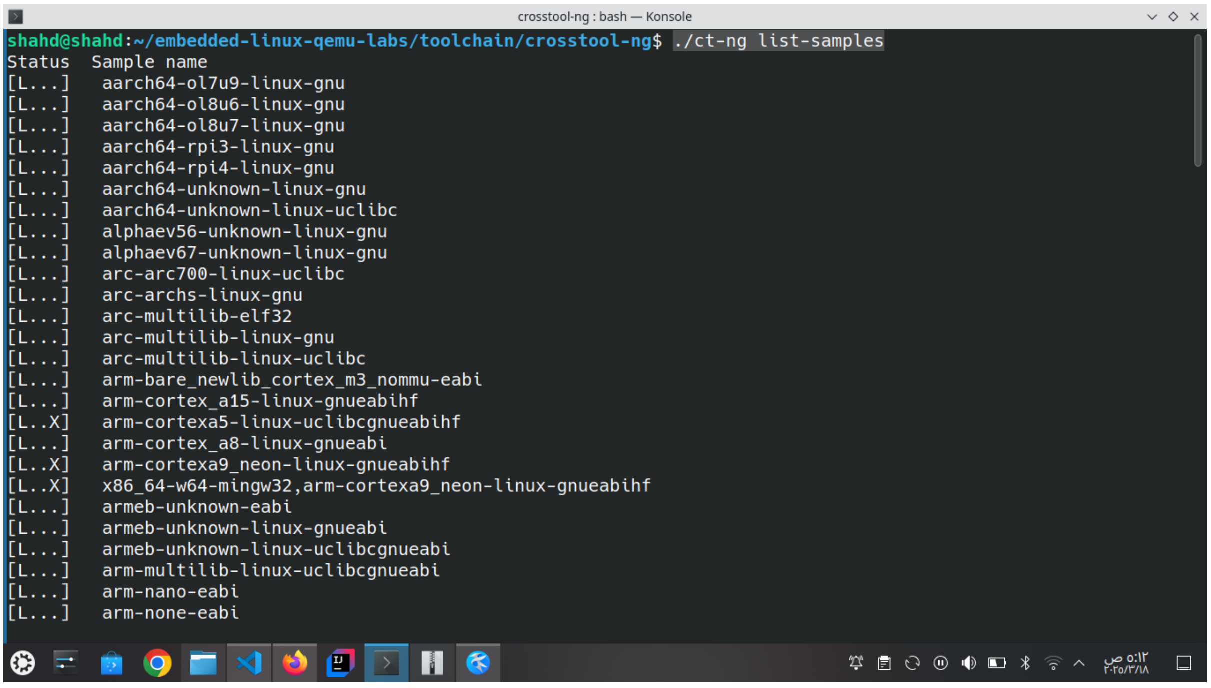
```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ ./bootstrap
INFO  :: *** Generating package version descriptions
INFO  :: Master packages: autoconf automake avr-libc binutils bison cloog dtc duma elf2flt expat gcc gdb gettex
t glibc gmp gnuprumcu isl libelf libiconv libtool linux ltrace m4 make mingw-w64 moxiebox mpc mpfr musl ncurses
 newlib-nano newlib picolibc strace uClibc-ng zlib zstd
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/cloog.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/gmp.in'
INFO  :: Generating 'config/versions/gnuprumcu.in'
INFO  :: Generating 'config/versions/isl.in'
INFO  :: Generating 'config/versions/libelf.in'
INFO  :: Generating 'config/versions/libiconv.in'
INFO  :: Generating 'config/versions/libtool.in'
INFO  :: Generating 'config/versions/linux.in'
INFO  :: Generating 'config/versions/ltrace.in'
INFO  :: Generating 'config/versions/m4.in'
```



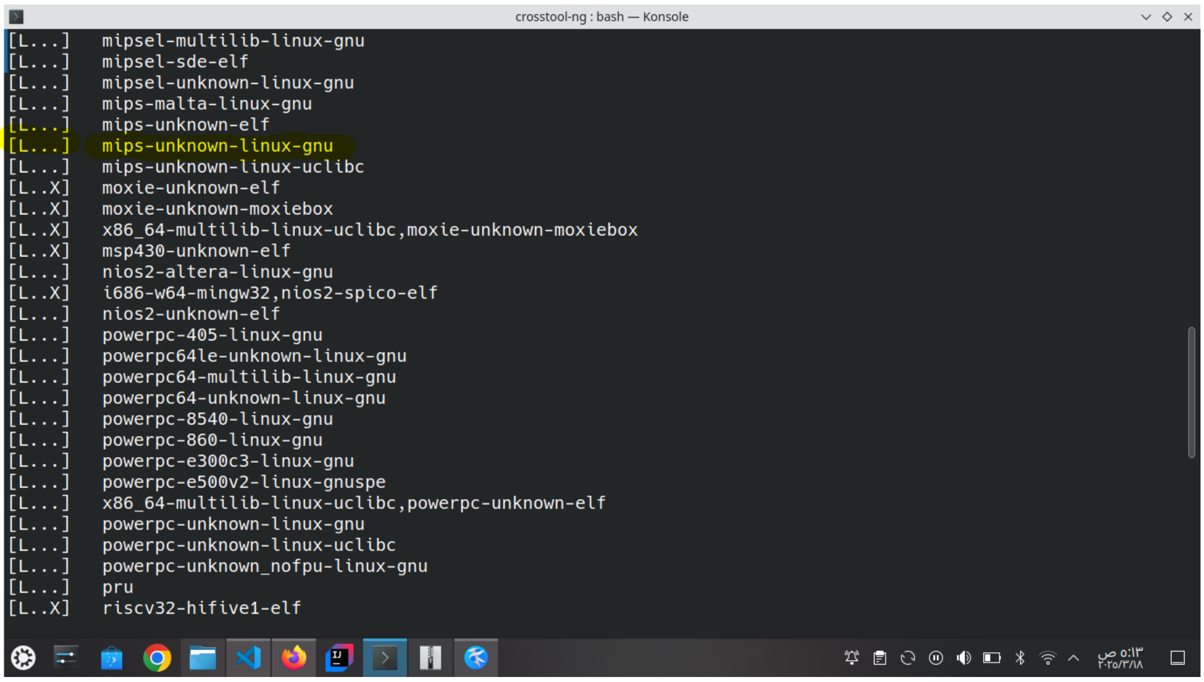
```
configure.ac:306: the top level
INFO  :: *** Done!
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ ./configure --enable-local
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking how to create a pax tar archive... gnutar
checking whether make supports nested variables... (cached) yes
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 >= 4.0... yes
checking for gcc... gcc
checking whether the C compiler works... yes
```



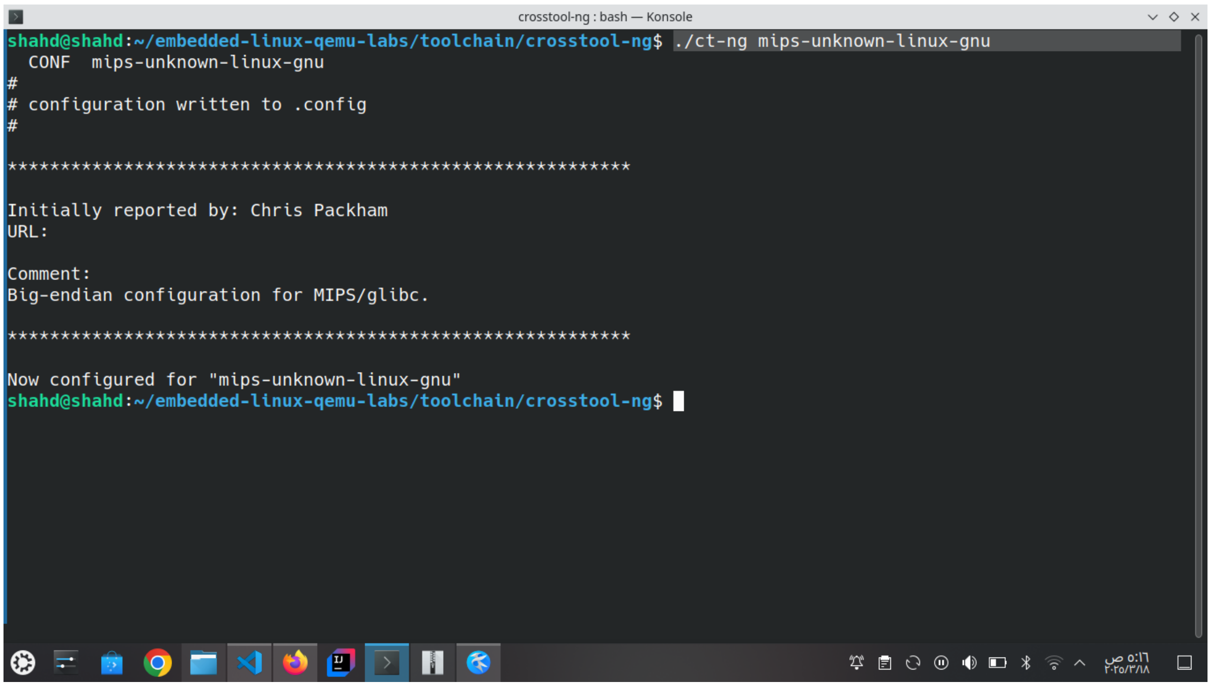
```
config.status: config.h is unchanged
config.status: executing depfiles commands
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ make
/usr/bin/gmake  all-recursive
gmake[1]: Entering directory '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng'
Making all in kconfig
gmake[2]: Entering directory '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng/kconfig'
/usr/bin/gmake  all-am
gmake[3]: Entering directory '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng/kconfig'
gmake[3]: Nothing to be done for 'all-am'.
gmake[3]: Leaving directory '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng/kconfig'
gmake[2]: Leaving directory '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng/kconfig'
gmake[2]: Entering directory '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng'
  GEN      ct-ng
  GEN      bash-completion/ct-ng
  GEN      docs/ct-ng.1
gmake[2]: Leaving directory '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng'
gmake[1]: Leaving directory '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng'
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ ./ct-ng help
This is crosstool-NG version 1.25.0.199_36ad0b1

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:
```



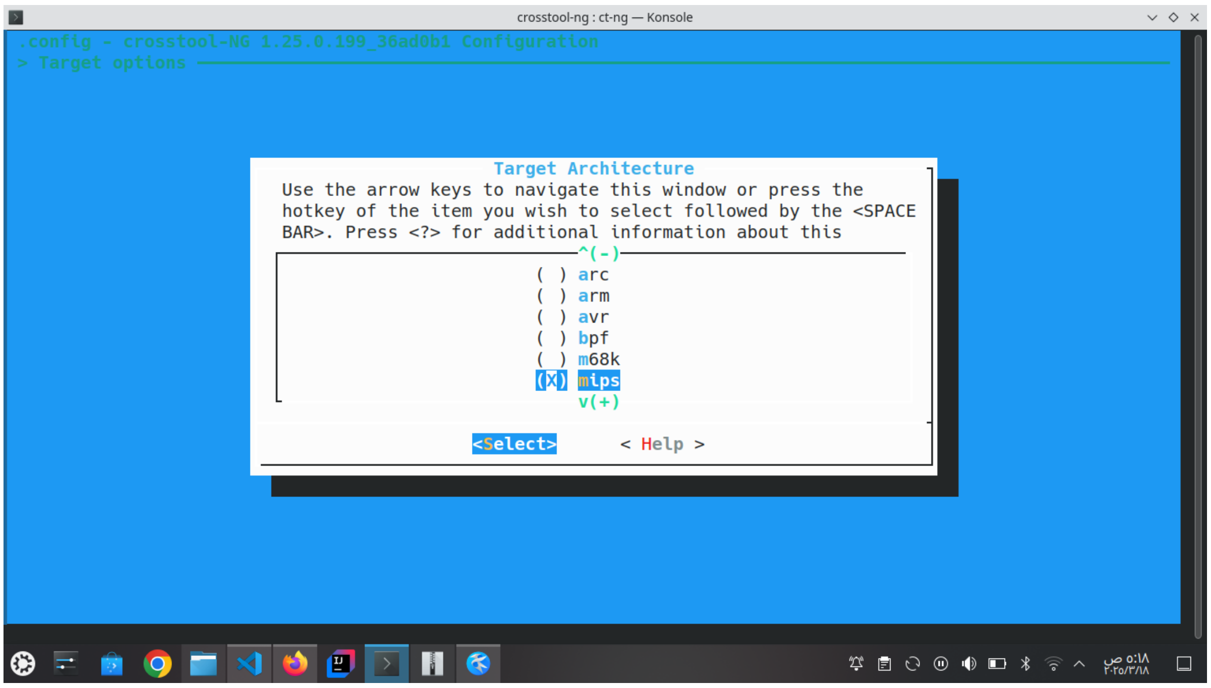
```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ ./ct-ng list-samples
Status  Sample name
[L...]   aarch64-ol7u9-linux-gnu
[L...]   aarch64-ol8u6-linux-gnu
[L...]   aarch64-ol8u7-linux-gnu
[L...]   aarch64-rpi3-linux-gnu
[L...]   aarch64-rpi4-linux-gnu
[L...]   aarch64-unknown-linux-gnu
[L...]   aarch64-unknown-linux-uclibc
[L...]   alphaev56-unknown-linux-gnu
[L...]   alphaev67-unknown-linux-gnu
[L...]   arc-arc700-linux-uclibc
[L...]   arc-archs-linux-gnu
[L...]   arc-multilib-elf32
[L...]   arc-multilib-linux-gnu
[L...]   arc-multilib-linux-uclibc
[L...]   arm-bare_newlib_cortex_m3_nommu-eabi
[L...]   arm-cortex_a15-linux-gnueabihf
[L..X]   arm-cortexa5-linux-uclibcgnueabihf
[L...]   arm-cortex_a8-linux-gnueabi
[L..X]   arm-cortexa9_neon-linux-gnueabihf
[L..X]   x86_64-w64-mingw32,arm-cortexa9_neon-linux-gnueabihf
[L...]   armeb-unknown-eabi
[L...]   armeb-unknown-linux-gnueabi
[L...]   armeb-unknown-linux-uclibcgnueabi
[L...]   arm-multilib-linux-uclibcgnueabi
[L...]   arm-nano-eabi
[L...]   arm-none-eabi
```



```
[L...]   mipsel-multilib-linux-gnu
[L...]   mipsel-sde-elf
[L...]   mipsel-unknown-linux-gnu
[L...]   mips-malta-linux-gnu
[L...]   mips-unknown-elf
[L...]   mips-unknown-linux-gnu
[L...]   mips-unknown-linux-uclibc
[L..X]   moxie-unknown-elf
[L..X]   moxie-unknown-moxiebox
[L..X]   x86_64-multilib-linux-uclibc,moxie-unknown-moxiebox
[L..X]   msp430-unknown-elf
[L...]   nios2-altera-linux-gnu
[L..X]   i686-w64-mingw32,nios2-spico-elf
[L...]   nios2-unknown-elf
[L...]   powerpc-405-linux-gnu
[L...]   powerpc64le-unknown-linux-gnu
[L...]   powerpc64-multilib-linux-gnu
[L...]   powerpc64-unknown-linux-gnu
[L...]   powerpc-8540-linux-gnu
[L...]   powerpc-860-linux-gnu
[L...]   powerpc-e300c3-linux-gnu
[L...]   powerpc-e500v2-linux-gnuspe
[L...]   x86_64-multilib-linux-uclibc,powerpc-unknown-elf
[L...]   powerpc-unknown-linux-gnu
[L...]   powerpc-unknown-linux-uclibc
[L...]   powerpc-unknown_nofpu-linux-gnu
[L...]   pru
[L..X]   riscv32-hifive1-elf
```

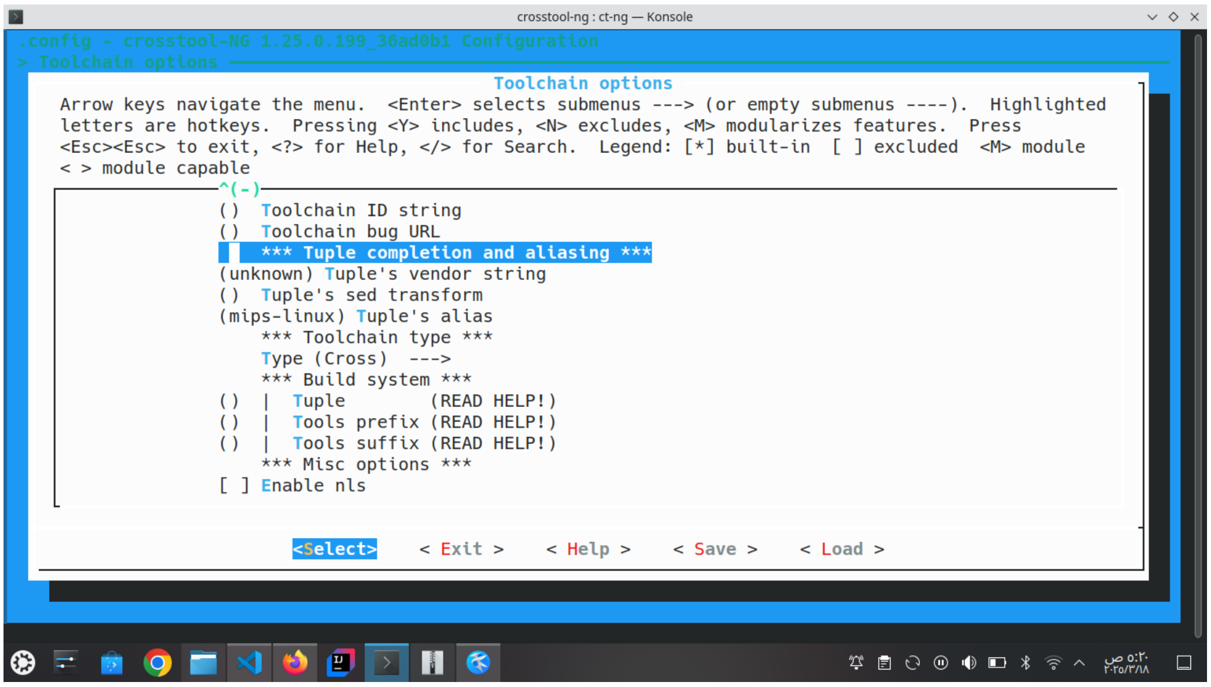


```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ ./ct-ng mips-unknown-linux-gnu
  CONF  mips-unknown-linux-gnu
#
# configuration written to .config
#

***********************************************************

Initially reported by: Chris Packham
URL:

Comment:
Big-endian configuration for MIPS/glibc.

***********************************************************

Now configured for "mips-unknown-linux-gnu"
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$
```



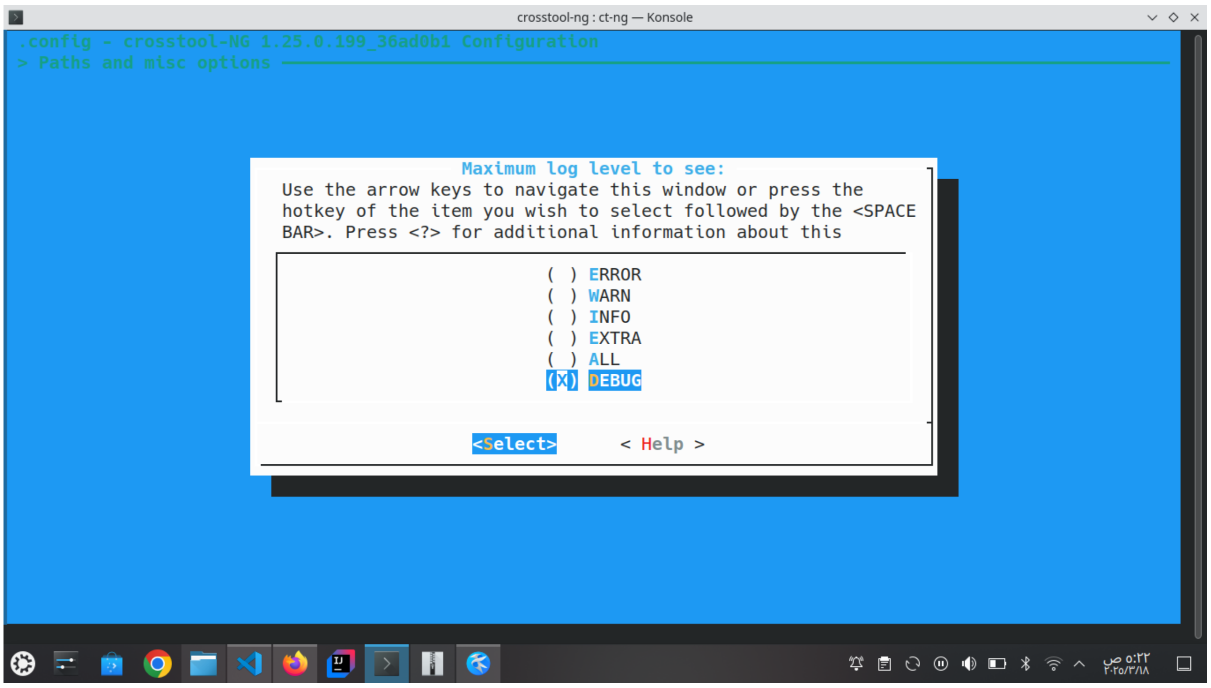
crosstool-ng : ct-ng — Konsole

```
.config - crosstool-NG 1.25.0.199_36ad0b1 Configuration
> Target options ─────────────────────────────────────────────────

            ┌──────────────── Target Architecture ────────────────┐
            │  Use the arrow keys to navigate this window or press the
            │  hotkey of the item you wish to select followed by the <SPACE
            │  BAR>. Press <?> for additional information about this
            │ ┌────────────────────────^(-)─────────────────────────┐
            │ │                       ( ) arc                        │
            │ │                       ( ) arm                        │
            │ │                       ( ) avr                        │
            │ │                       ( ) bpf                        │
            │ │                       ( ) m68k                       │
            │ │                       (X) mips                       │
            │ └─────────────────────────v(+)───────────────────────┘
            │                 <Select>      < Help >                │
            └──────────────────────────────────────────────────────┘
```

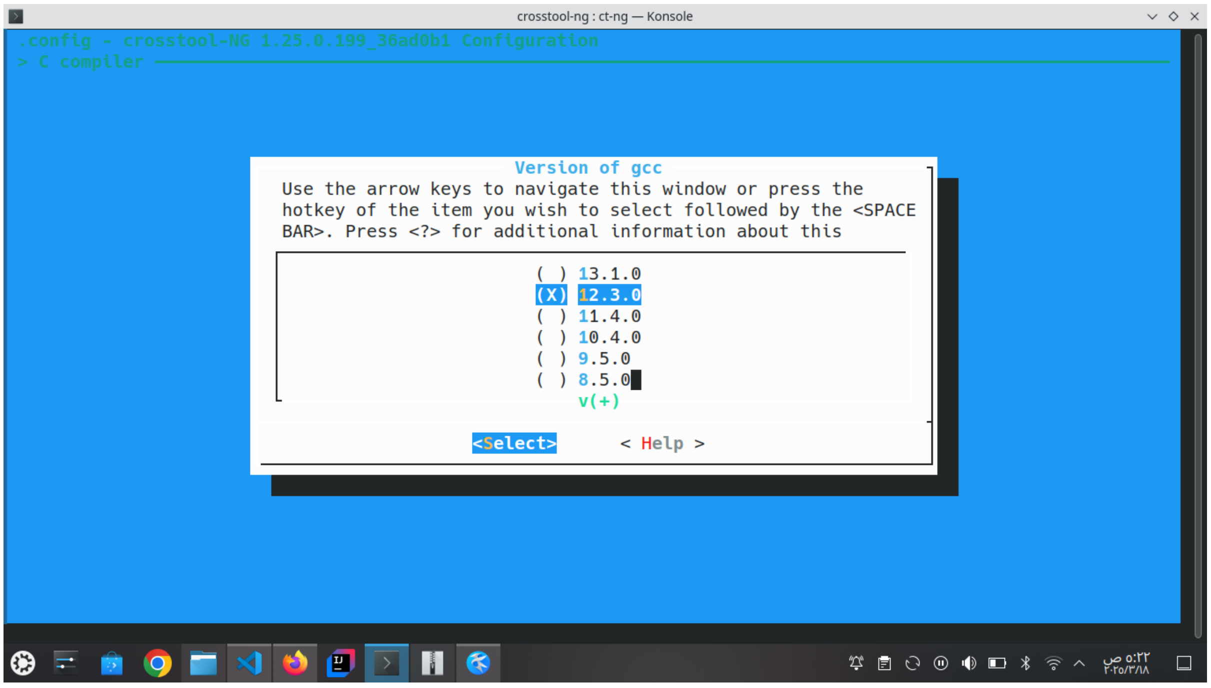


crosstool-ng : ct-ng — Konsole

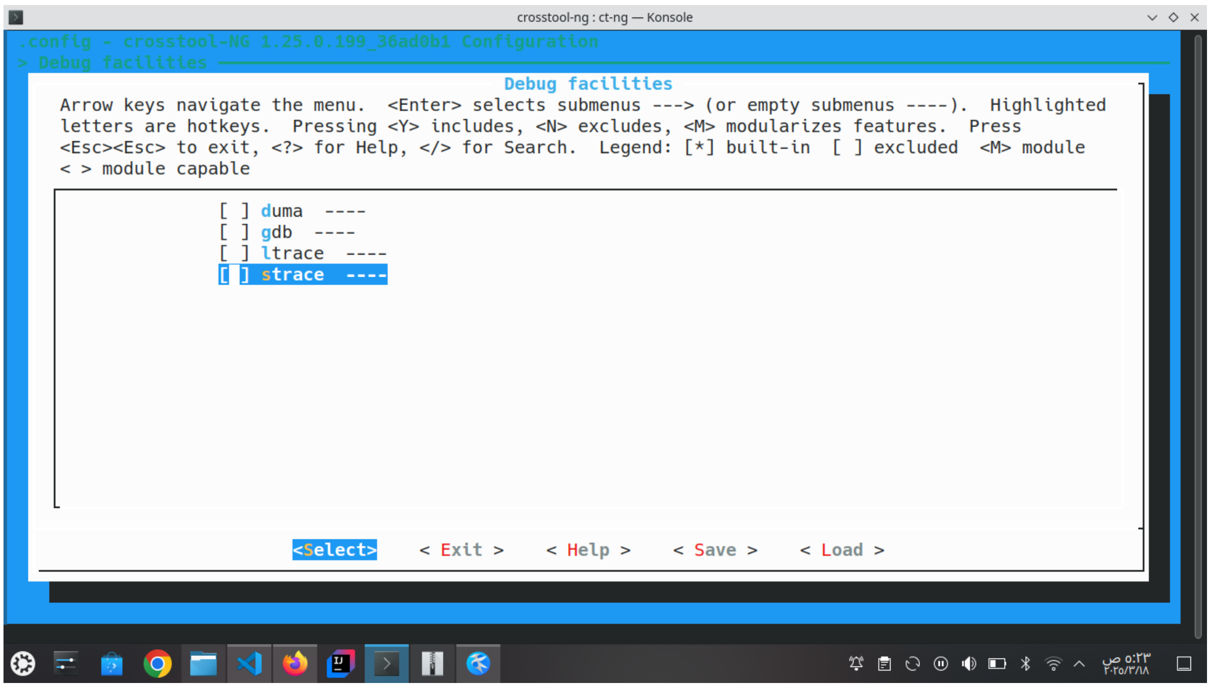
```
.config - crosstool-NG 1.25.0.199_36ad0b1 Configuration
> Toolchain options ──────────────────────────────────────────────
 ┌───────────────────────── Toolchain options ─────────────────────────┐
 │  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
 │  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
 │  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module
 │  < > module capable
 │ ┌───────────────^(-)────────────────────────────────────────────┐
 │ │               ()  Toolchain ID string
 │ │               ()  Toolchain bug URL
 │ │                   *** Tuple completion and aliasing ***
 │ │               (unknown) Tuple's vendor string
 │ │               ()  Tuple's sed transform
 │ │               (mips-linux) Tuple's alias
 │ │                   *** Toolchain type ***
 │ │                   Type (Cross)  --->
 │ │                   *** Build system ***
 │ │               ()  |  Tuple        (READ HELP!)
 │ │               ()  |  Tools prefix (READ HELP!)
 │ │               ()  |  Tools suffix (READ HELP!)
 │ │                   *** Misc options ***
 │ │               [ ] Enable nls
 │ └───────────────────────────────────────────────────────────────┘
 │          <Select>    < Exit >    < Help >    < Save >    < Load >
 └─────────────────────────────────────────────────────────────────────┘
```



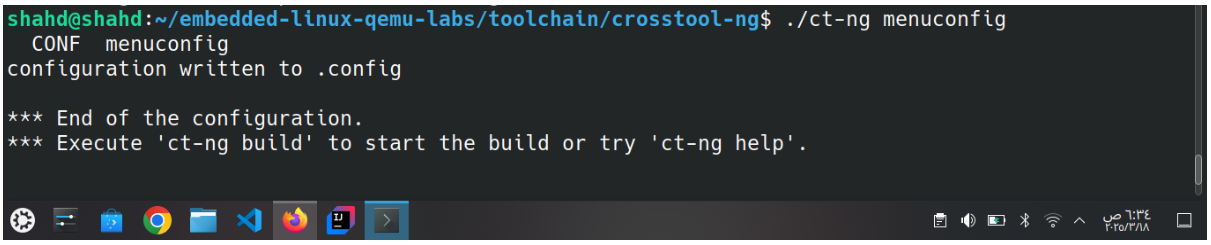
```
crosstool-ng : ct-ng — Konsole
.config - crosstool-NG 1.25.0.199_36ad0b1 Configuration
> Paths and misc options ──────────────────────────────────────

                    Maximum log level to see:
 Use the arrow keys to navigate this window or press the
 hotkey of the item you wish to select followed by the <SPACE
 BAR>. Press <?> for additional information about this

                         ( ) ERROR
                         ( ) WARN
                         ( ) INFO
                         ( ) EXTRA
                         ( ) ALL
                         (X) DEBUG

              <Select>      < Help >
```



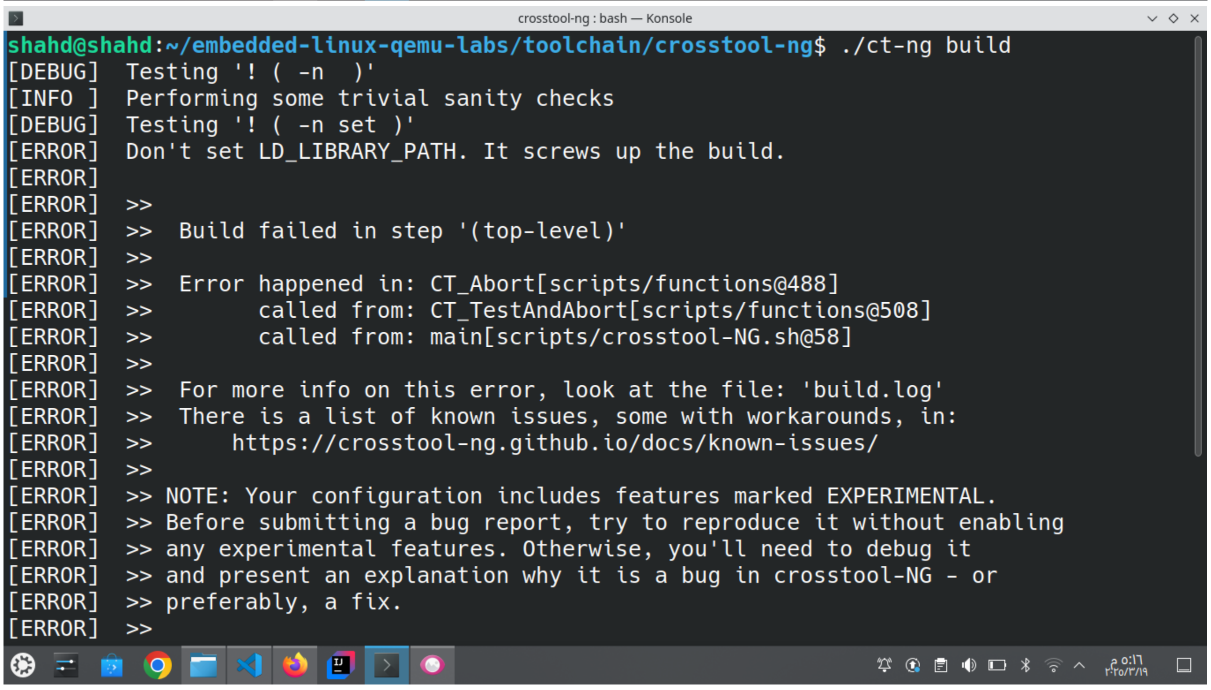
```
crosstool-ng : ct-ng — Konsole
.config - crosstool-NG 1.25.0.199_36ad0b1 Configuration
> C compiler ──────────────────────────────────────────────────

                        Version of gcc
 Use the arrow keys to navigate this window or press the
 hotkey of the item you wish to select followed by the <SPACE
 BAR>. Press <?> for additional information about this

                        ( ) 13.1.0
                        (X) 12.3.0
                        ( ) 11.4.0
                        ( ) 10.4.0
                        ( ) 9.5.0
                        ( ) 8.5.0
                            v(+)

              <Select>      < Help >
```



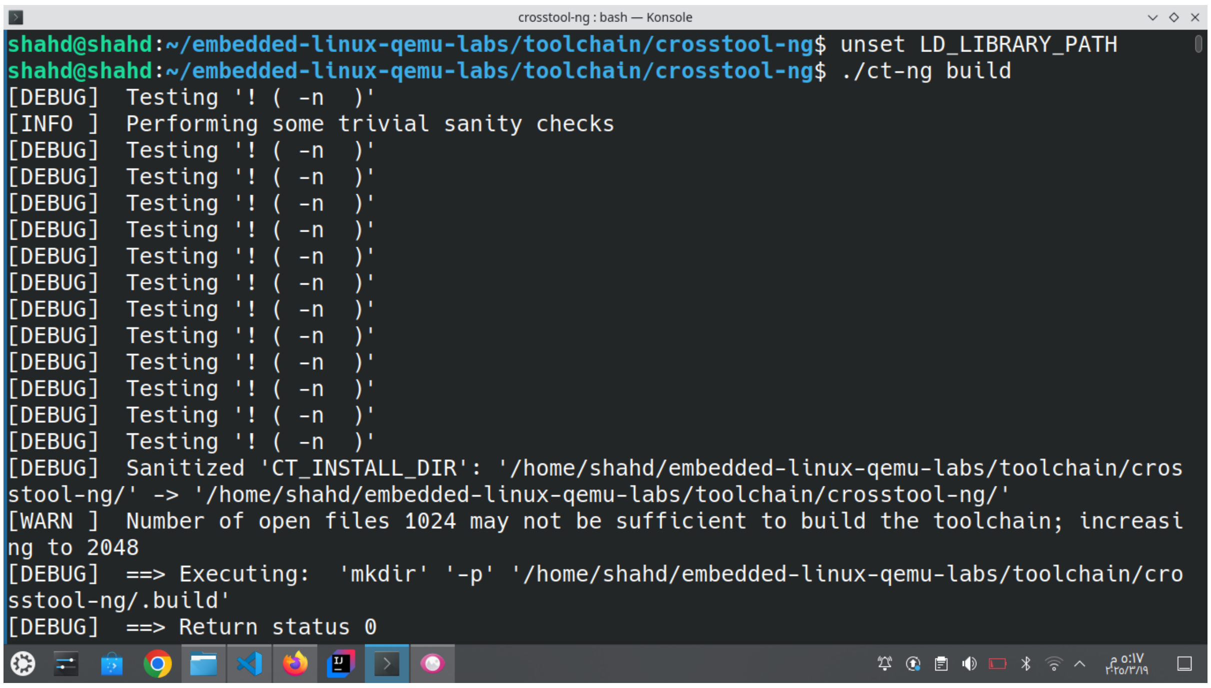
```
.config - crosstool-NG 1.25.0.199_36ad0b1 Configuration
> Debug facilities ──────────────────────────────────────────
                              Debug facilities
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module
  < > module capable

                [ ] duma  ----
                [ ] gdb  ----
                [ ] ltrace  ----
                [ ] strace  ----

                <Select>    < Exit >    < Help >    < Save >    < Load >
```



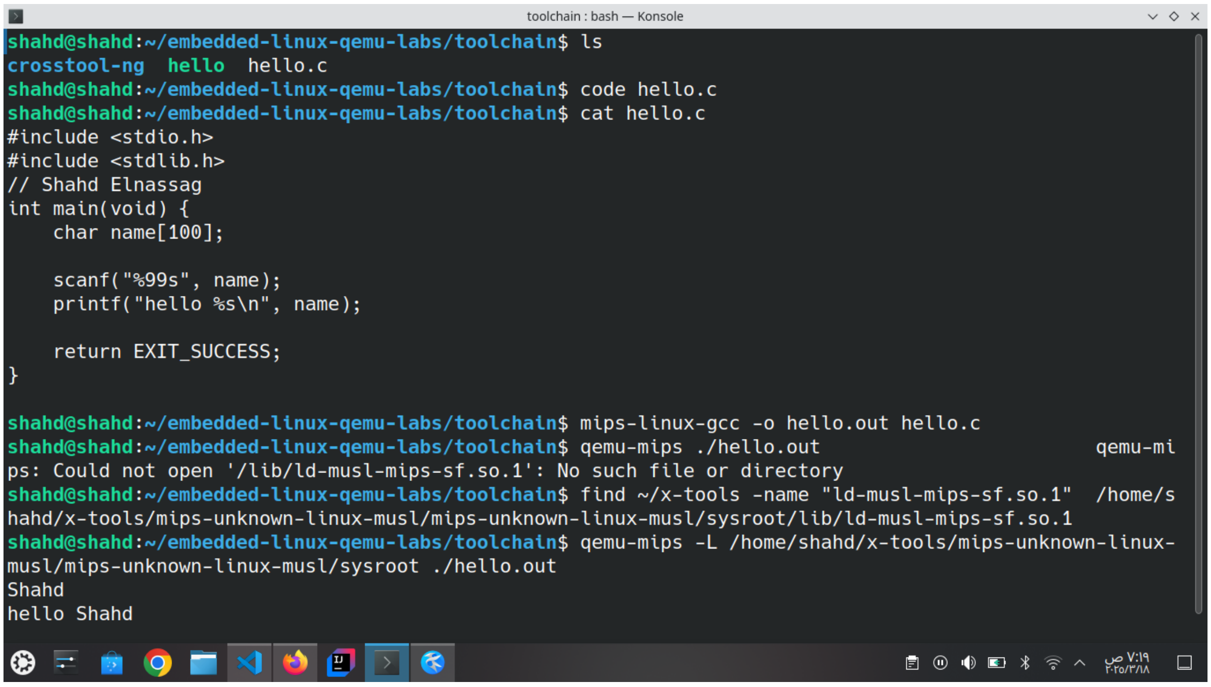
```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ ./ct-ng menuconfig
  CONF  menuconfig
configuration written to .config

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



```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ ./ct-ng build
[DEBUG]  Testing '! ( -n  )'
[INFO ]  Performing some trivial sanity checks
[DEBUG]  Testing '! ( -n set )'
[ERROR]  Don't set LD_LIBRARY_PATH. It screws up the build.
[ERROR]
[ERROR]  >>
[ERROR]  >>  Build failed in step '(top-level)'
[ERROR]  >>
[ERROR]  >>  Error happened in: CT_Abort[scripts/functions@488]
[ERROR]  >>        called from: CT_TestAndAbort[scripts/functions@508]
[ERROR]  >>        called from: main[scripts/crosstool-NG.sh@58]
[ERROR]  >>
[ERROR]  >>  For more info on this error, look at the file: 'build.log'
[ERROR]  >>  There is a list of known issues, some with workarounds, in:
[ERROR]  >>      https://crosstool-ng.github.io/docs/known-issues/
[ERROR]  >>
[ERROR]  >> NOTE: Your configuration includes features marked EXPERIMENTAL.
[ERROR]  >> Before submitting a bug report, try to reproduce it without enabling
[ERROR]  >> any experimental features. Otherwise, you'll need to debug it
[ERROR]  >> and present an explanation why it is a bug in crosstool-NG - or
[ERROR]  >> preferably, a fix.
[ERROR]  >>
```



```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ unset LD_LIBRARY_PATH
shahd@shahd:~/embedded-linux-qemu-labs/toolchain/crosstool-ng$ ./ct-ng build
[DEBUG]  Testing '! ( -n  )'
[INFO ]  Performing some trivial sanity checks
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Testing '! ( -n  )'
[DEBUG]  Sanitized 'CT_INSTALL_DIR': '/home/shahd/embedded-linux-qemu-labs/toolchain/cros
stool-ng/' -> '/home/shahd/embedded-linux-qemu-labs/toolchain/crosstool-ng/'
[WARN ]  Number of open files 1024 may not be sufficient to build the toolchain; increasi
ng to 2048
[DEBUG]  ==> Executing:  'mkdir' '-p' '/home/shahd/embedded-linux-qemu-labs/toolchain/cro
sstool-ng/.build'
[DEBUG]  ==> Return status 0
```



```
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ ls
crosstool-ng  hello  hello.c
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ code hello.c
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ cat hello.c
#include <stdio.h>
#include <stdlib.h>
// Shahd Elnassag
int main(void) {
    char name[100];

    scanf("%99s", name);
    printf("hello %s\n", name);

    return EXIT_SUCCESS;
}

shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ mips-linux-gcc -o hello.out hello.c
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ qemu-mips ./hello.out                   qemu-mi
ps: Could not open '/lib/ld-musl-mips-sf.so.1': No such file or directory
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ find ~/x-tools -name "ld-musl-mips-sf.so.1"  /home/s
hahd/x-tools/mips-unknown-linux-musl/mips-unknown-linux-musl/sysroot/lib/ld-musl-mips-sf.so.1
shahd@shahd:~/embedded-linux-qemu-labs/toolchain$ qemu-mips -L /home/shahd/x-tools/mips-unknown-linux-
musl/mips-unknown-linux-musl/sysroot ./hello.out
Shahd
hello Shahd
```