
PRACTICAL EXAM RTES

LEVEL 0

Baseline



PACKAGES INSTALLATION

First

Check which of the following packages can be installed without downloading them from the internet.

```
bitbake -s | less
```

strace
socat
vim
gawk

```
1: iot@iot-VirtualBox: ~/WORKDIR/Yocto/poky
ssh-pregen-hostkeys :1.0-r0
startup-notification :0.12-r2
strace :5.17-r0
strace-native :5.17-r0

1: iot@iot-VirtualBox: ~/WORKDIR/Yocto/poky
vala-native :0.56.1-r0
valgrind :3.19.0-r0
vim :8.2.4912-r0
vim-native :8.2.4912-r0

1: iot@iot-VirtualBox: ~/WORKDIR/Yocto/poky/build
virgslang :2.3.2-r0
slang-native :2.3.2-r0
socat :1.7.4.3-r0
socat-native :1.7.4.3-r0

1: iot@iot-VirtualBox: ~/WORKDIR/Yocto/poky/build
fribidi :1.0.12-r0
fribidi-native :1.0.12-r0
gawk :5.1.1-r0
gawk-native :5.1.1-r0
gcc :12.1.0-r0
gcc-cross-aarch64 :12.1.0-r0
```

PACKAGES INSTALLATION

Add Packages

Add all the packages in the *local.conf* file using the *IMAGE_INSTALL* keyword and the *:append* particle.

Doing so they will be installed.

build/conf/local.conf

```
...  
IMAGE_INSTALL:append = " strace"  
IMAGE_INSTALL:append = " socat"  
IMAGE_INSTALL:append = " vim"  
IMAGE_INSTALL:append = " gawk"  
...
```

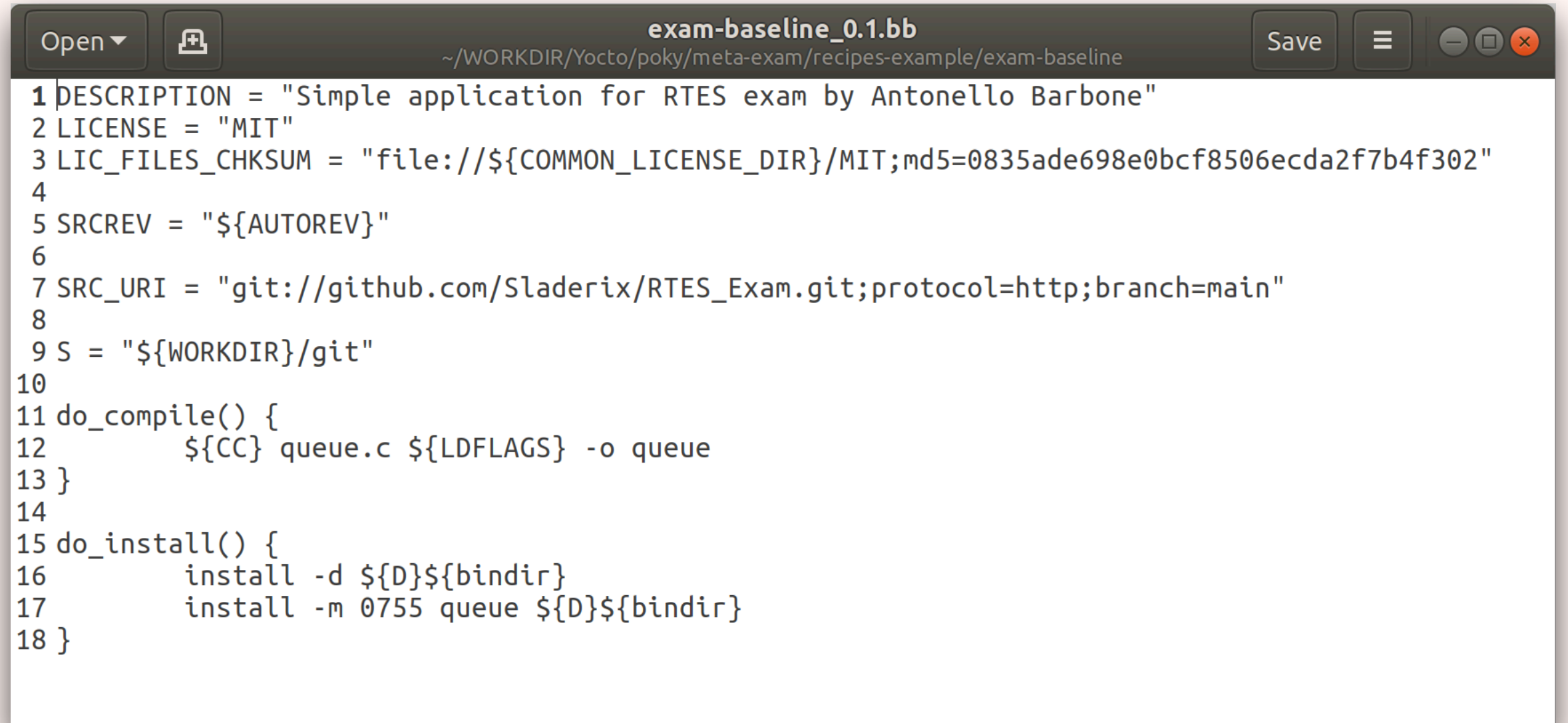
CUSTOM APPLICATION

Coding

The custom application is coded in C and then uploaded to a [GitHub repository](#).

Yocto Recipe

The recipe is configured to download the repository and take the C file from there.

A screenshot of a code editor window showing a Yocto recipe file named 'exam-baseline_0.1.bb'. The window has a dark title bar with 'exam-baseline_0.1.bb' and a path '~/.WORKDIR/Yocto/poky/meta-exam/recipes-example/exam-baseline'. There are 'Open', 'Save', and window control buttons. The code is a BitBake recipe for a simple application. It includes a description, license, source URI, and compile/install functions.

```
1 DESCRIPTION = "Simple application for RTES exam by Antonello Barbone"
2 LICENSE = "MIT"
3 LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"
4
5 SRCREV = "${AUTOREV}"
6
7 SRC_URI = "git://github.com/Sladerix/RTES_Exam.git;protocol=http;branch=main"
8
9 S = "${WORKDIR}/git"
10
11 do_compile() {
12     ${CC} queue.c ${LDFLAGS} -o queue
13 }
14
15 do_install() {
16     install -d ${D}${bindir}
17     install -m 0755 queue ${D}${bindir}
18 }
```

LEVEL 1

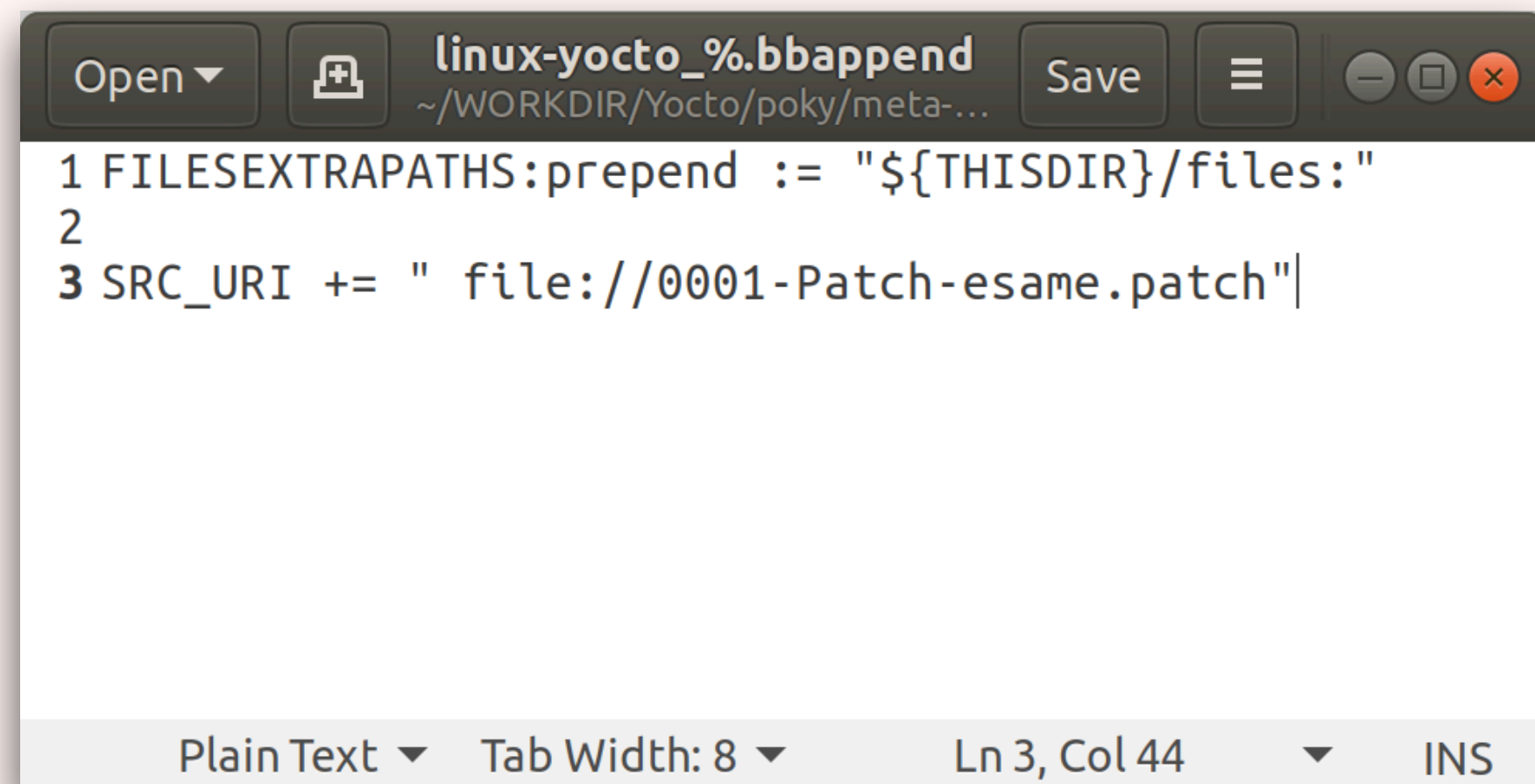
Advanced



PATCHING

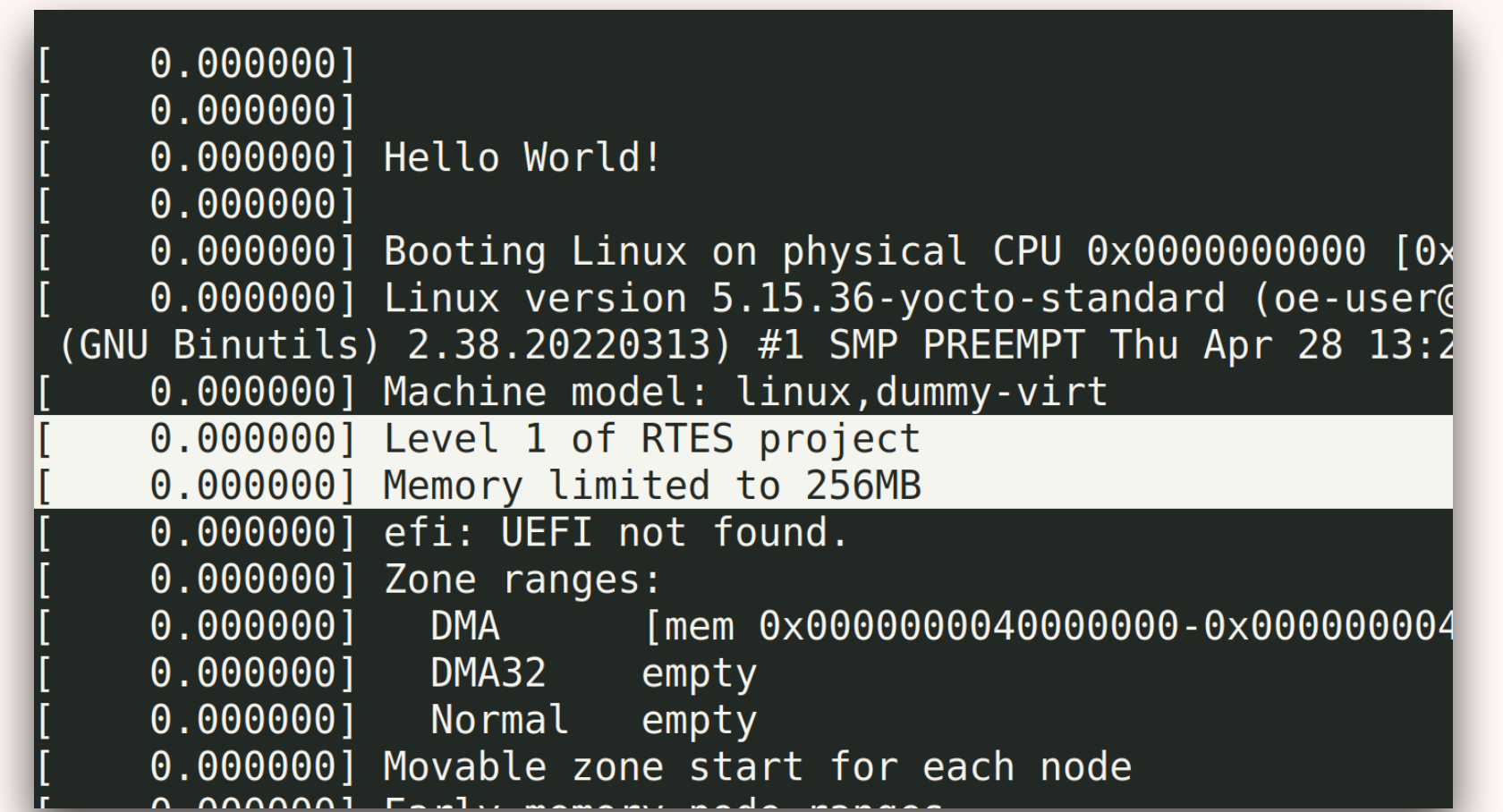
Add the Patch

In the `meta/meta-exam/` layer folder, created for the exam, was create also the `recipe-kernel/linux/` folder needed to contain the patches for the kernel. In this last folder is contained the `.bbappend` file that specifies where to get the patch file (E.g. contained in the `files/` folder).



```
1 FILESEXTRAPATHS:prepend := "${THISDIR}/files:"  
2  
3 SRC_URI += " file://0001-Patch-esame.patch"
```

The screenshot shows a text editor window titled "linux-yocto_%.bbappend" with a file path starting with "~/.WORKDIR/Yocto/poky/meta-...". The editor contains three lines of text: a line defining FILESEXTRAPATHS:prepend, a blank line, and a line defining SRC_URI to include a patch file. The status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 3, Col 44", and "INS" mode.



```
[ 0.000000]  
[ 0.000000]  
[ 0.000000] Hello World!  
[ 0.000000]  
[ 0.000000] Booting Linux on physical CPU 0x000000000000 [0x  
[ 0.000000] Linux version 5.15.36-yocto-standard (oe-user@  
(GNU Binutils) 2.38.20220313) #1 SMP PREEMPT Thu Apr 28 13:2  
[ 0.000000] Machine model: linux,dummy-virt  
[ 0.000000] Level 1 of RTES project  
[ 0.000000] Memory limited to 256MB  
[ 0.000000] efi: UEFI not found.  
[ 0.000000] Zone ranges:  
[ 0.000000]   DMA      [mem 0x00000000040000000-0x0000000004  
[ 0.000000]   DMA32    empty  
[ 0.000000]   Normal   empty  
[ 0.000000] Movable zone start for each node  
[ 0.000000] Early memory node ranges
```

The screenshot shows a terminal window with boot logs. The logs include messages about booting Linux, the kernel version (5.15.36-yocto-standard), machine model (linux,dummy-virt), memory limits (256MB), and zone ranges (DMA, DMA32, Normal). The logs are displayed in a dark-themed terminal window.

LEVEL 2

Expert



THREAD APPLICATION

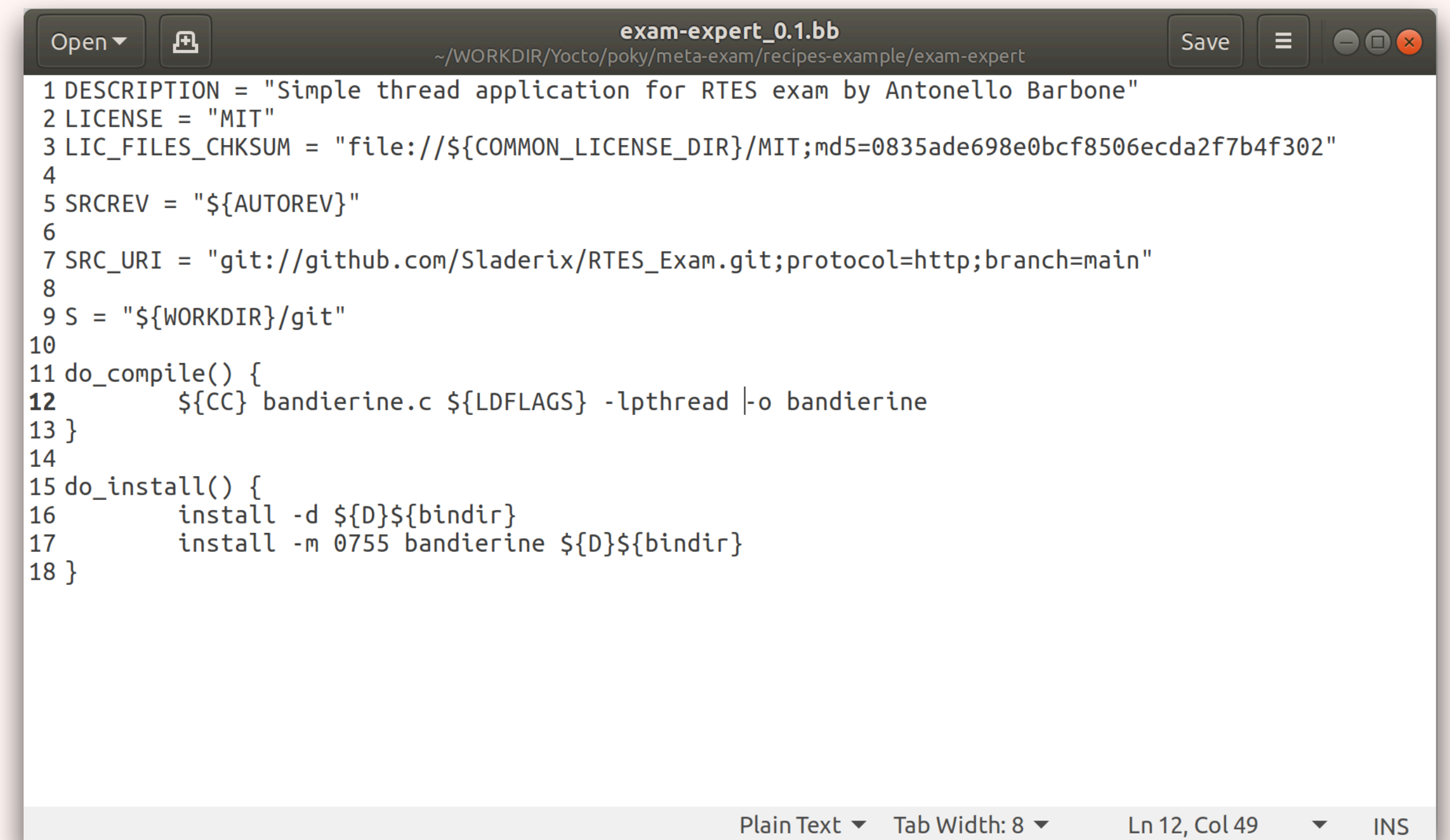
Coding

The thread application is coded in C and then uploaded to a [GitHub repository](#).

Yocto Recipe

The recipe is configured to download the repository and take the C file from there.

The flag *-lpthread* is specified to the cross-compiler.



```
1 DESCRIPTION = "Simple thread application for RTE exam by Antonello Barbone"
2 LICENSE = "MIT"
3 LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"
4
5 SRCREV = "${AUTOREV}"
6
7 SRC_URI = "git://github.com/Sladerix/RTE_Exam.git;protocol=http;branch=main"
8
9 S = "${WORKDIR}/git"
10
11 do_compile() {
12     ${CC} bandierine.c ${LDFLAGS} -lpthread -o bandierine
13 }
14
15 do_install() {
16     install -d ${D}${bindir}
17     install -m 0755 bandierine ${D}${bindir}
18 }
```

The screenshot shows a text editor window titled 'exam-expert_0.1.bb' with a file path of '~/.WORKDIR/Yocto/poky/meta-exam/recipes-example/exam-expert'. The editor contains a Yocto recipe with 18 lines of code. The code defines a simple thread application, sets the license to MIT, specifies the source URI as a GitHub repository, and defines the compile and install tasks. The compile task uses the C compiler and the pthread library to create an executable named 'bandierine'. The install task creates a directory and installs the executable with permissions 0755. The status bar at the bottom indicates 'Plain Text', 'Tab Width: 8', 'Ln 12, Col 49', and 'INS'.