
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 ▾

vim-vim:8.2.4912-r0

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"  
...
```

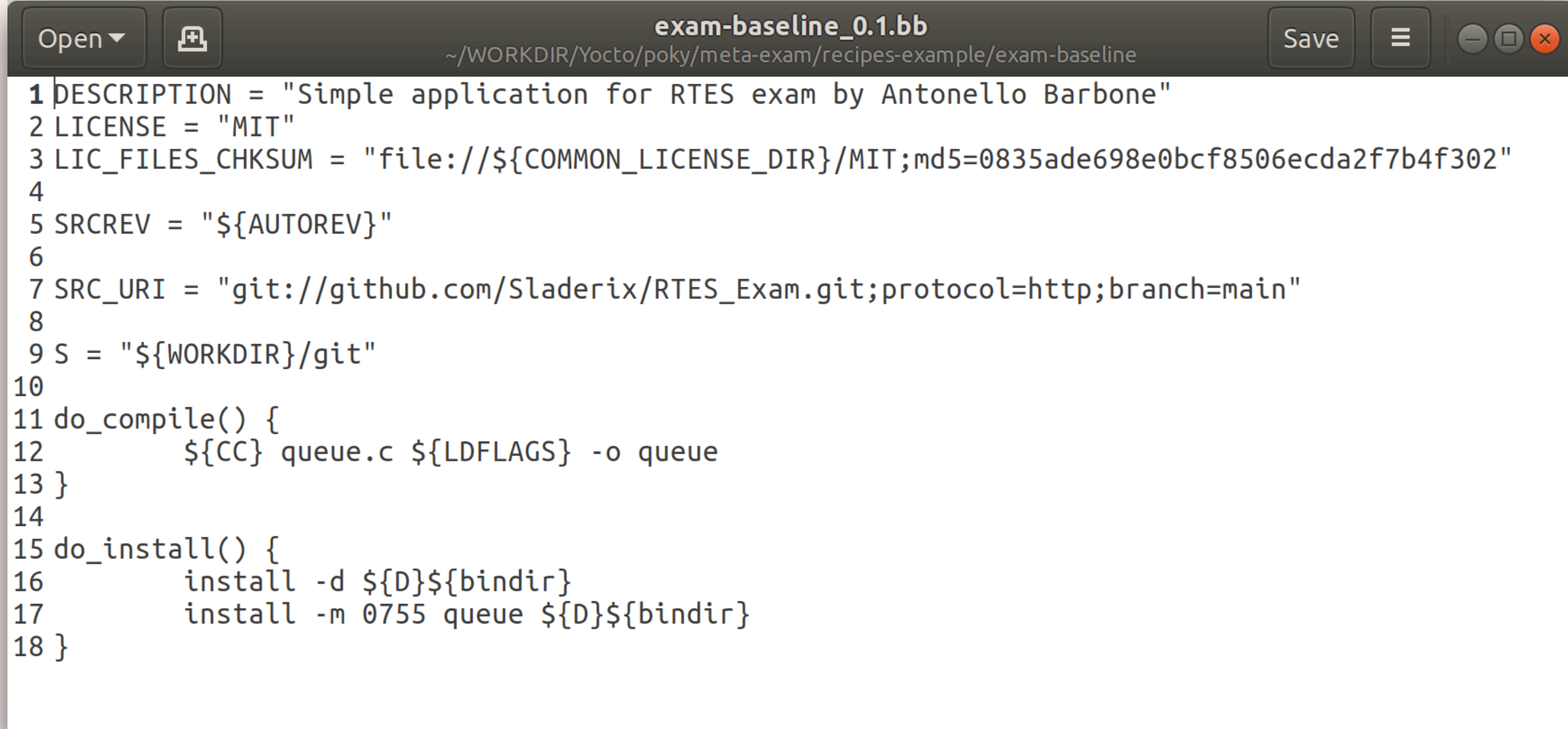
CUSTOM APPLICATION

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 text 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'. The file content is as follows:

```
1 DESCRIPTION = "Simple 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} 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

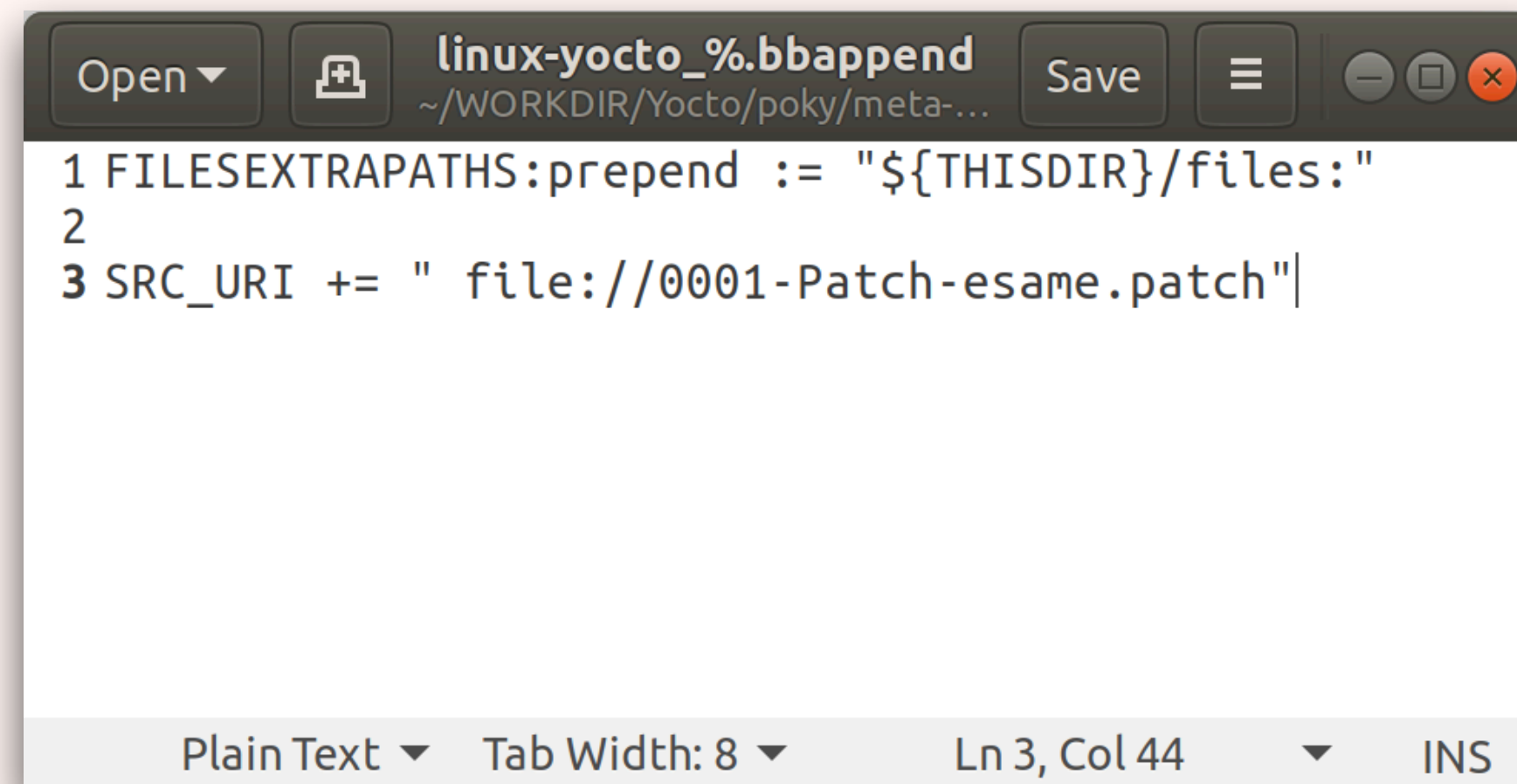
Advanced



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).



The screenshot shows a text editor window titled "linux-yocto_%.bbappend" with a path of "~/WORKDIR/Yocto/poky/meta-...". The editor contains three lines of text: "1 FILESEXTRAPATHS:prepend := "\${THISDIR}/files:", "2", and "3 SRC_URI += " file://0001-Patch-esame.patch"|". The status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 3, Col 44", and "INS".

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

LEVEL 2

Expert



THREAD APPLICATION

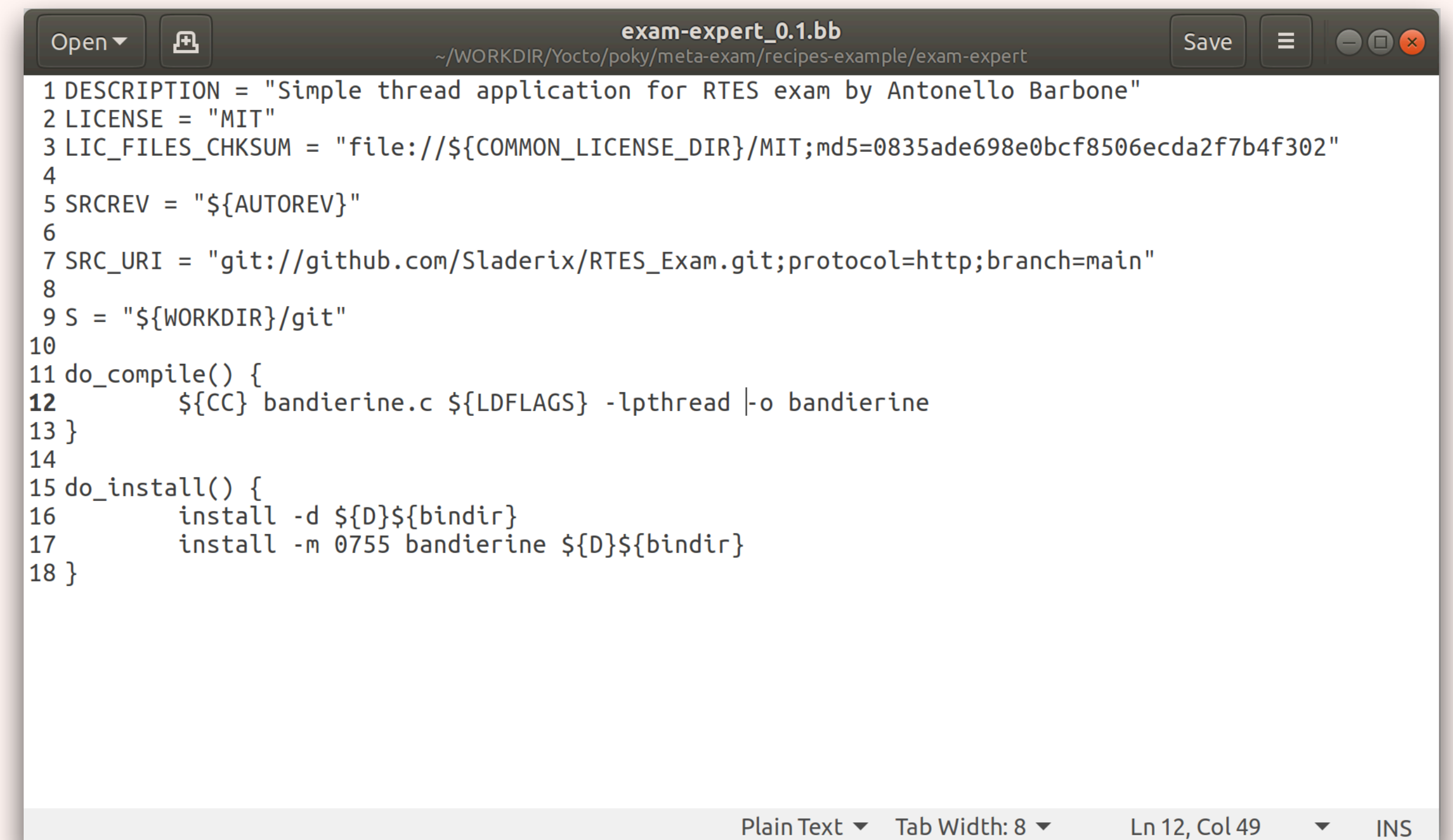
Coding

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 }
```