**PRACTICA Introducción al uso del lenguaje Kconfig**

**Introducción**

Esta práctica está diseñada para ejemplificar una especie de programa “hola mundo” en el lenguaje Kconfig.

Instale el paquete kconfig-frontends-nox (para instalar el ejecutable kconfig-mconf).

$ sudo apt-get install kconfig-frontends-nox

**Desarrollo**

Para el desarrollo de esta práctica se utilizarán tres archivos de código fuente en lenguaje C, un archivo Kconfig, y un archivo make. El archivo Kconfig, se usará a través del objetivo menuconfig del archivo make, para realizar la configuración de un proyecto. El proyecto se realizará en Codeblocks (aunque podría usarse algún otro IDE que utilice el toolchain de GNU). Los listados de los archivos se muestran a continuación.

Archivo Ejemplo\_1.c

#include <stdio.h> /\* comment \*/

#ifdef CONFIG\_EJEMPLO\_\_\_I

int main(void)

{

printf("Hello\n");

printf("Welcome to the Course!\n");

return 0;

}/\*end main()\*/

#endif

Archivo Ejemplo\_2.c

#include <stdio.h> /\* comment \*/

#ifdef CONFIG\_EJEMPLO\_\_II

int main(void)

{

int a,b;

printf("Enter two numbers: ");

scanf("%i %i",&a,&b);

printf("%i - %i = %i\n",a,b,a-b);

return 0;

}/\*end main()\*/

#endif

Archivo Ejemplo\_3.c

#include <stdio.h>

#include <limits.h>

#ifdef CONFIG\_EJEMPLO\_III

int main(void)

{

unsigned long big = ULONG\_MAX;

printf("minimum int = %i, ",INT\_MIN);

printf("maximum int = %i\n",INT\_MAX);

printf("maximum unsigned = %u\n",UINT\_MAX);

printf("maximum long int = %li\n",LONG\_MAX);

printf("maximum unsigned long = %lu\n",big);

return 0;

}

#endif

Nótese que los tres archivos anteriores contienen una función main(), pero las directivas (#ifdef, #endif) se usarán para realizar compilación condicional. La configuración del proyecto se realizará a través del siguiente archivo Kconfig

Archivo Kconfig

config EJEMPLO\_\_\_I

bool "Ejemplo 1 programa 'Hola Mundo'"

default y

help

Se selecciona el programa del ejemplo 1 (diapositiva 8)

config EJEMPLO\_\_II

bool "Ejemplo 2 programa que usa las funciones printf y scanf"

default n

help

Se selecciona el programa del ejemplo 2 (diapositiva 11)

config EJEMPLO\_III

bool "Ejemplo 3 programa que muestra el especificador de formato %li (diapositiva 20)"

default n

help

Se selecciona el programa del ejemplo 3. El programa incluye limits.h con lo que tiene

acceso a las macros ULONG\_MAX INT\_MIN INT\_MAX UINT\_MAX LONG\_MAX.

El archivo make utiliza kconfig-mconf para interpretar el archivo Kconfig cuando se ejecuta el comando

$ make menuconfig

Archivo Makefile

default:

@echo "Use make menuconfig para configurar el proyecto"

menuconfig:

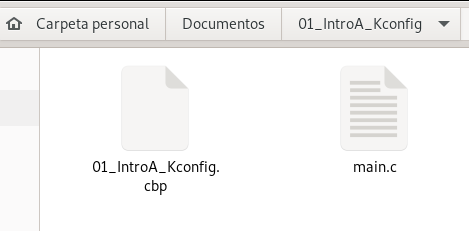
kconfig-mconf Kconfig

./scripts/kconfig/conf --silentoldconfig Kconfig

rm -v obj/Release/\*.o

**Modificación del contenido del directorio del proyecto de Codeblocks**

Cuando se crea el proyecto 01\_IntroA\_Kconfig en codeblocks, esta aplicación nos crea un directorio cuyo nombre es el mismo que el nombre del proyecto, y dentro de este directorio, un archivo con el nombre del proyecto y extensión cbp (code blocks project), como se muestra en la siguiente figura.



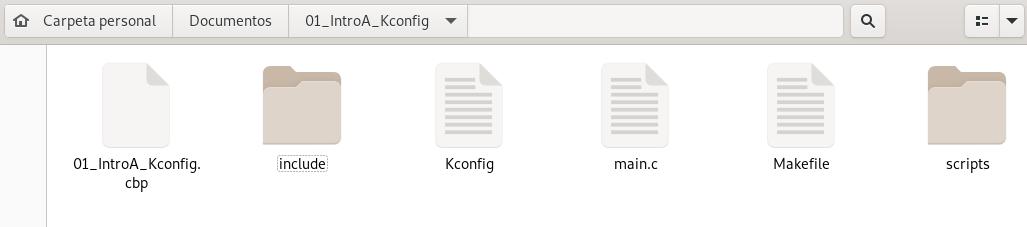
Para los propósitos de esta práctica, en el directorio del proyecto 01\_IntroA\_Kconfig/, deberemos copiar el archivo Makefile y los directorios include/ y scripts/, que se pueden descargar desde la url (respectivamente):

https://github.com/progavan/2021-2/tree/main/UNIDAD00/01\_IntroA\_Kconfig/Makefile

https://github.com/progavan/2021-2/tree/main/UNIDAD00/01\_IntroA\_Kconfig/include

https://github.com/progavan/2021-2/tree/main/UNIDAD00/01\_IntroA\_Kconfig/scripts.tar

para que nuestro directorio local 01\_IntroA\_Kconfig/, nos quede como se muestra a continuación



o equivalentemente: (En el directorio include debemos colocar dos subdirectorios: config y generated)









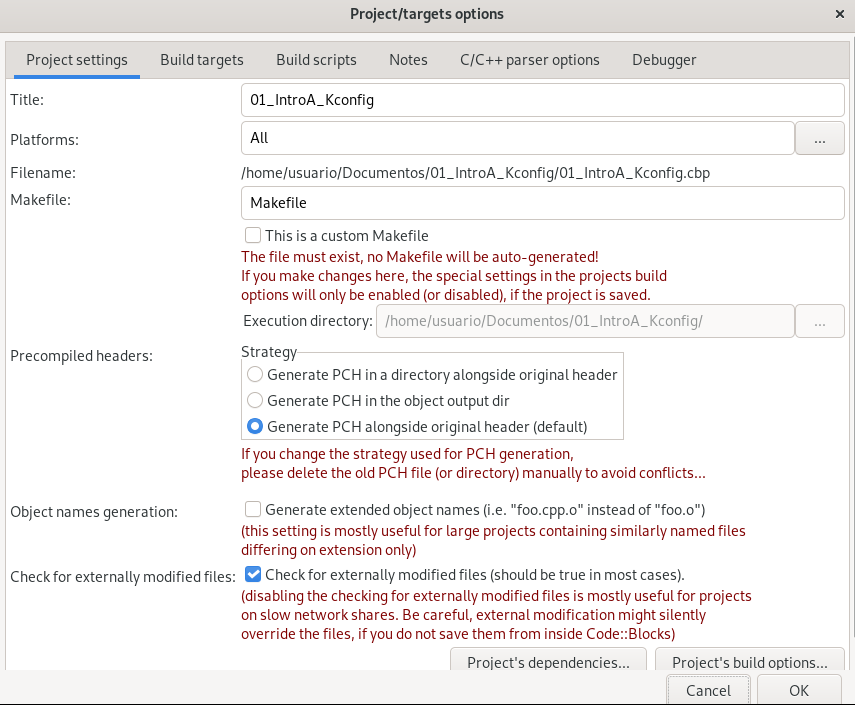
Necesitamos el directorio scripts/, porque en el objetivo menuconfig, del archivo make, se utilizará el archivo ejecutable scripts/kconfig/conf

Por otra parte, los directorios include/ e include/generated/ se utilizarán para crear y actualizar el archivo autoconf.h

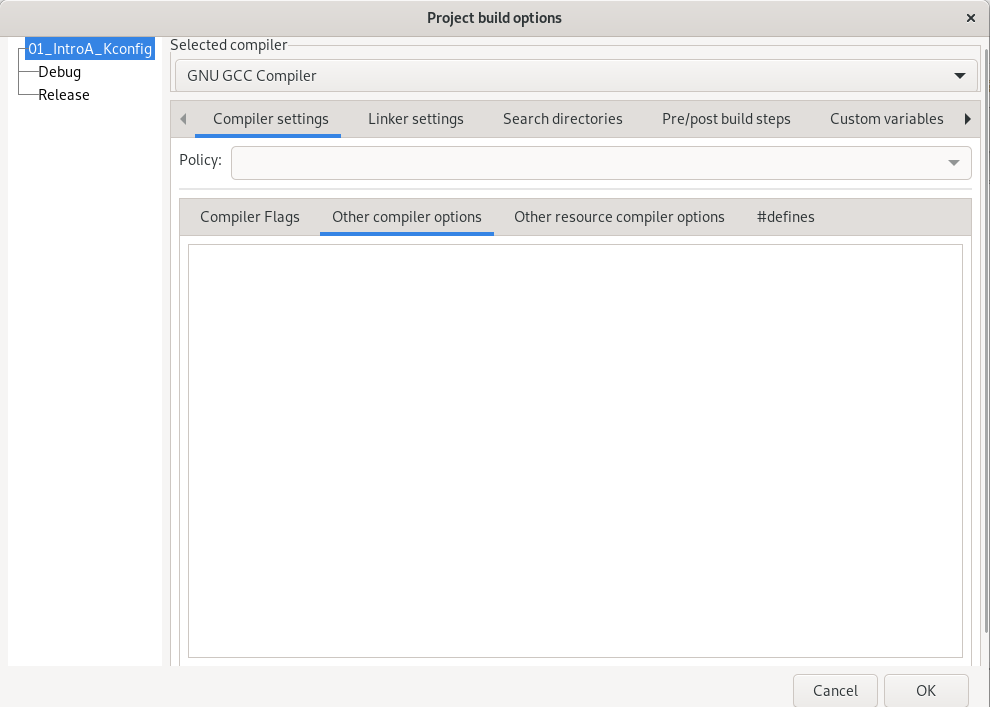
**Modificación en la configuración del proyecto de codeblocks**

Se accederá al dialogo para agregar configuración al proyecto como sigue:

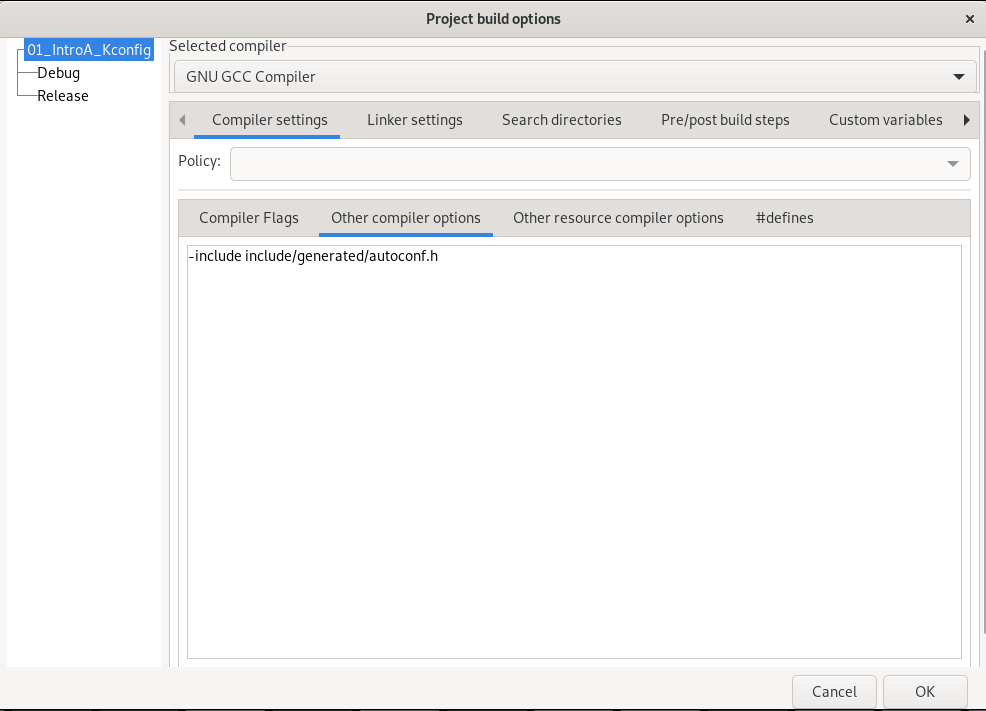
Project – > Properties – > project’s Build Options <enter> (véase esquina inferior derecha de la siguiente figura)



Dar clic en “Other compiler options”

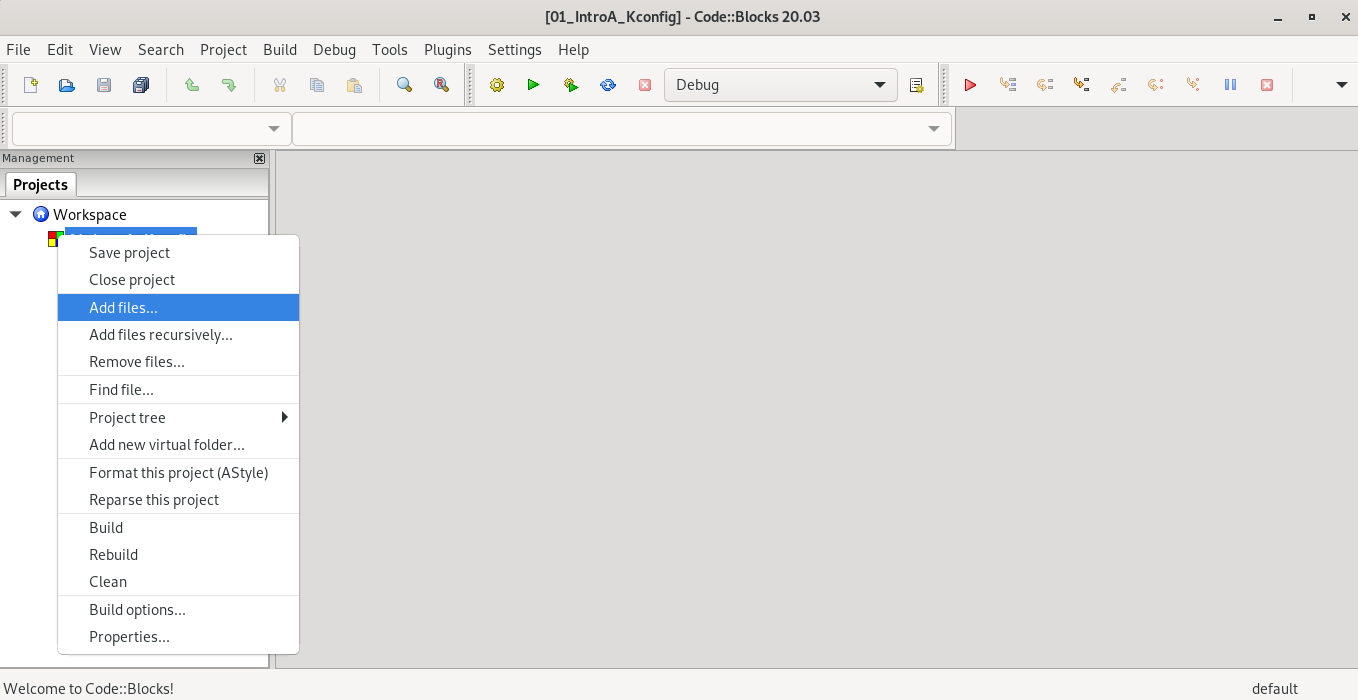


y agregamos la opción -include include/generated/autoconf.h

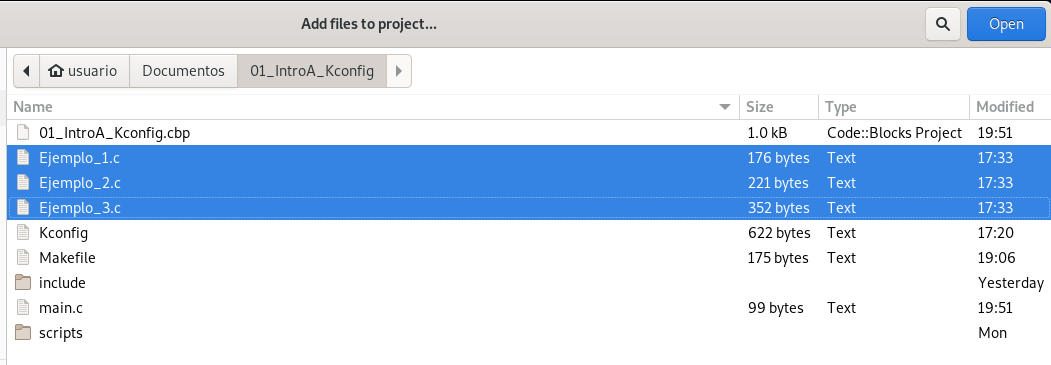


damos clic en el botón OK.

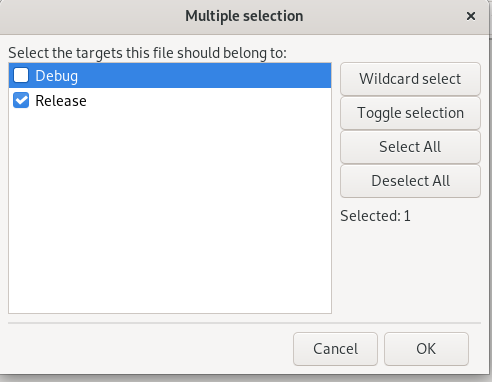
En este punto, podemos agregar los archivos fuente Ejemplo\_1.c, Ejemplo\_2.c, y Ejemplo\_3.c al proyecto de codeblocks.



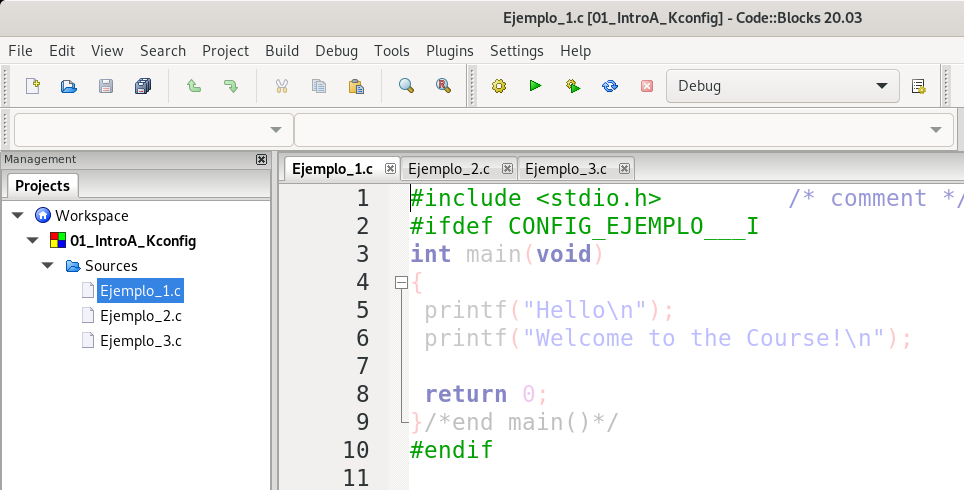
Y seleccionamos los tres archivos mencionados



Los agregaremos al objetivo Release



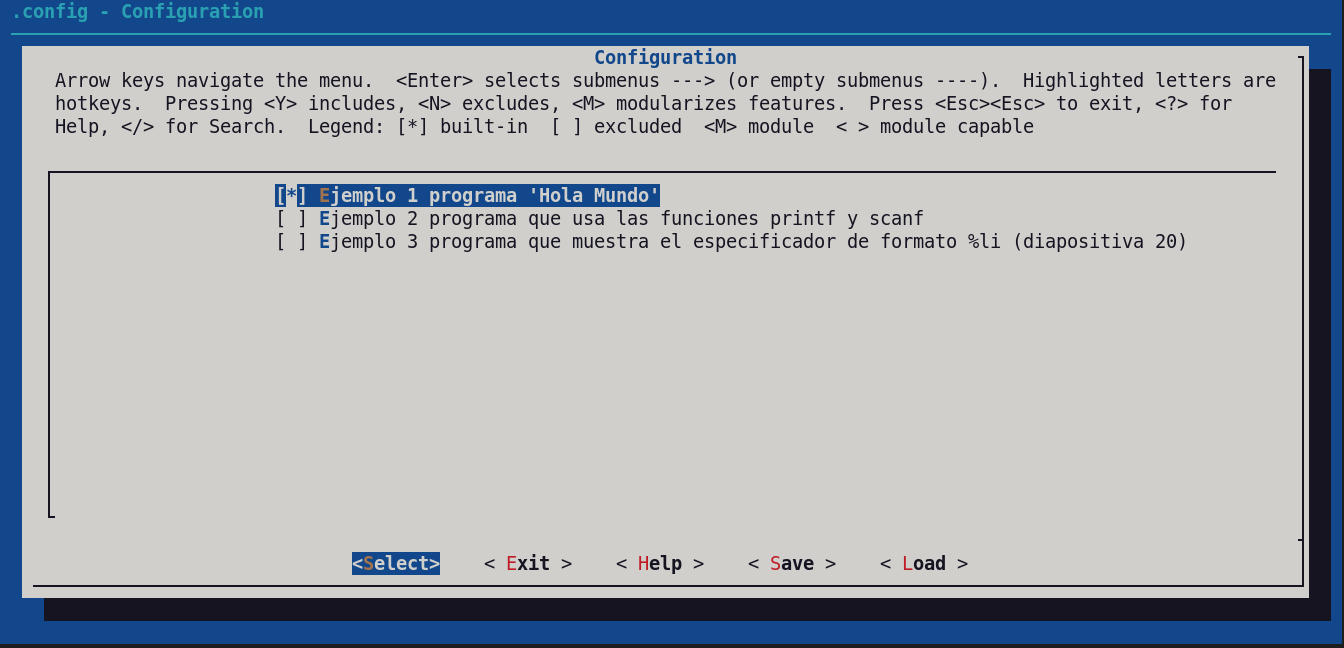
En el panel de proyectos, se mostrarán los archivos que acabamos de agregar



(Note que el archivo main.c que originalmente nos proporciona codeblocks, ha sido removido del proyecto).

**Configuración del proyecto (Ejecución/interpretación del archivo Kconfig)**

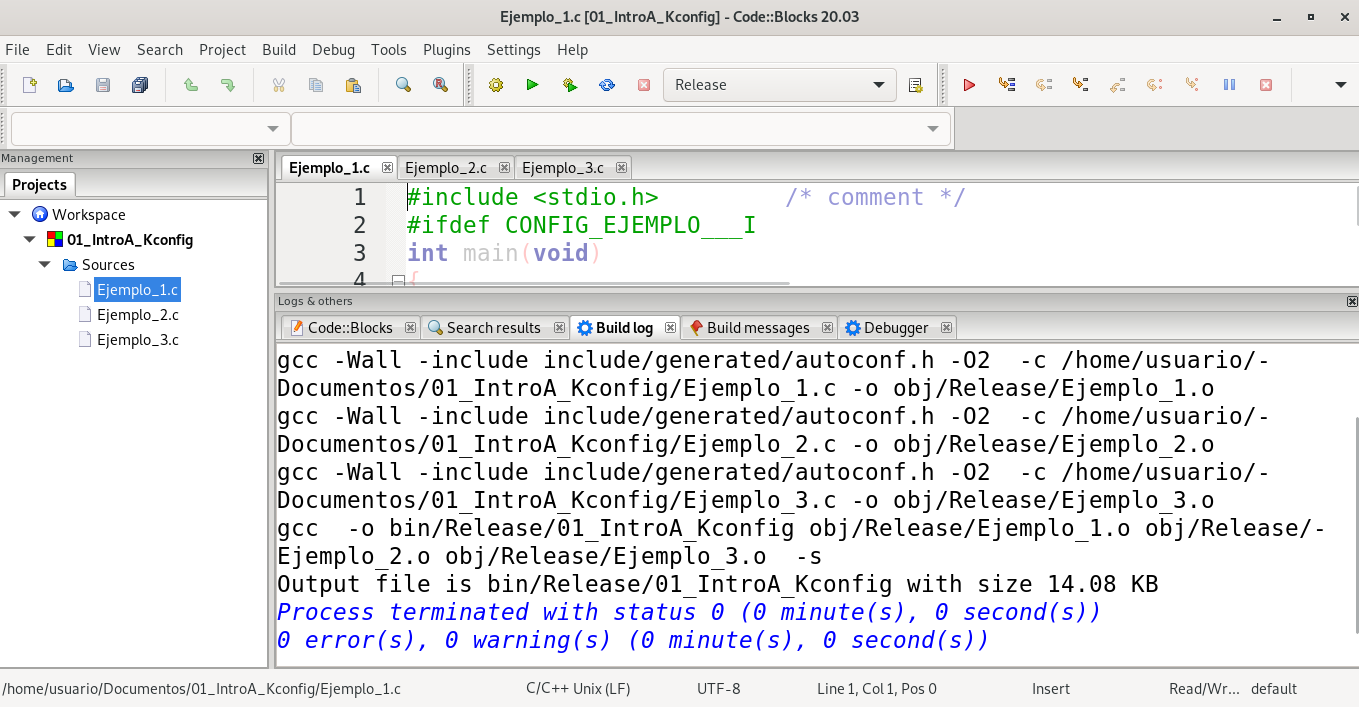
A continuación, podemos configurar el proyecto en una terminal abierta en el directorio base del proyecto ejecutando make menuconfig.



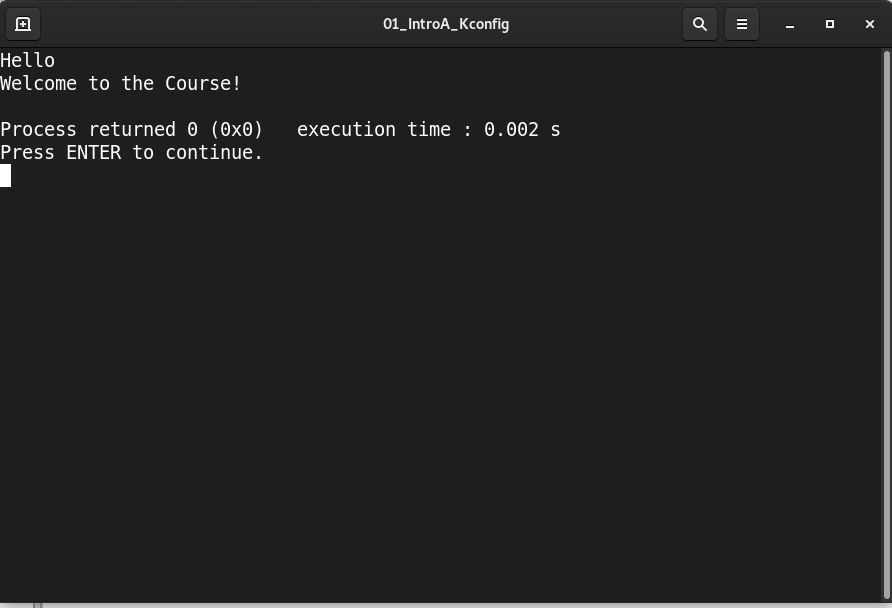
e dialogo, debemos seleccionar solo una de las tres opciones, y después guardar la configuración , seleccionando sucesivamente la opción <Exit>.

**Construcción y ejecución del programa seleccionado en la configuración**

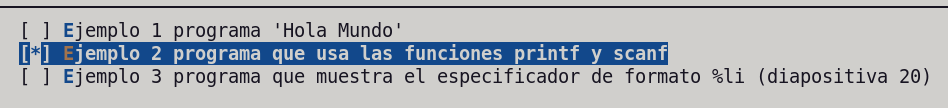
Con todos los pasos anteriores, ya debemos poder construir exitosamente en codeblocks, el archivo ejecutable correspondiente al archivo de código fuente seleccionado en el dialogo que se muestra en la terminal cuando ejecutamos el comando make menuconfig. Un ejemplo de construcción se muestra en la siguiente figura.



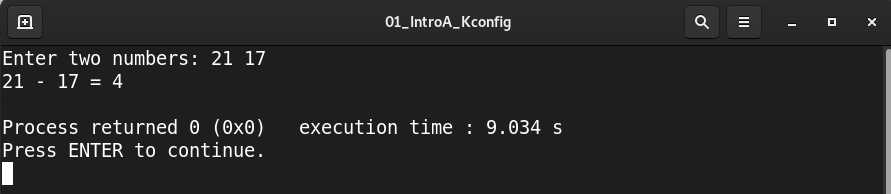
En la siguiente figura, se muestra la ejecución del ejecutable construido a partir del archivo fuente Ejemplo\_1.c



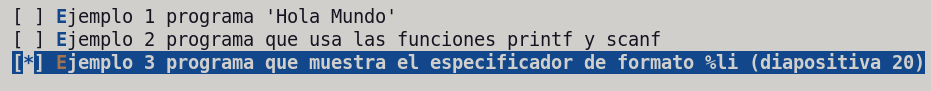
Para completar la práctica, haga nuevamente la configuración ejecutando make menuconfig en la terminal seleccionando



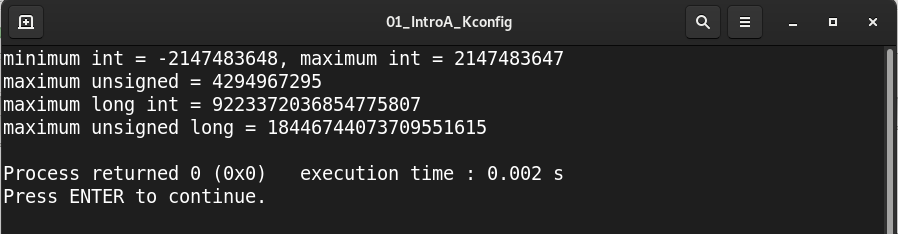
Construya y ejecute nuevamente el proyecto para obtener la salida del ejemplo 2.



Finalmente, reconfigure por tercera vez el proyecto seleccionando el ejemplo 3.



Construya y ejecute el proyecto para obtener la salida del ejemplo 3.



**Referencias**

https://boozlachu.medium.com/using-kconfig-for-own-projects-deb21e0d1804

Using\_kconfig\_for\_own\_projects.docx (Docto. Anexo).

Documentación de las fuentes del kernel linux 5.10