



Sistemas Operativos

*Importación de proyectos con
Makefile en Eclipse y perfiles de
ejecución/depuración*



Contenido



- Alternativas para el desarrollo de aplicaciones C en Linux
- Creación de proyecto vacío + Importación
- Generación de perfil de ejecución
- Generación de perfil de depuración
- Uso de GNU make para compilar desde el terminal



Desarrollo de aplicaciones C en Linux

- Tres alternativas
 1. Editor de Textos + GNU Make + Terminal
 2. Proyecto “C” con Makefile + IDE
 3. Proyecto “C” + IDE (compilación autogestionada)
- Eclipse es el IDE instalado en las distribuciones de GNU/Linux del laboratorio
- Para el desarrollo de las prácticas de esta asignatura usaremos las dos primeras alternativas (basadas en Make)
 - Con cada práctica y programa de ejemplo se incluirá un Makefile para la compilación del proyecto asociado
 - Ventaja: según convenga podemos cambiar entre el IDE y el terminal para ejecutar el programa o depurarlo de distintas formas



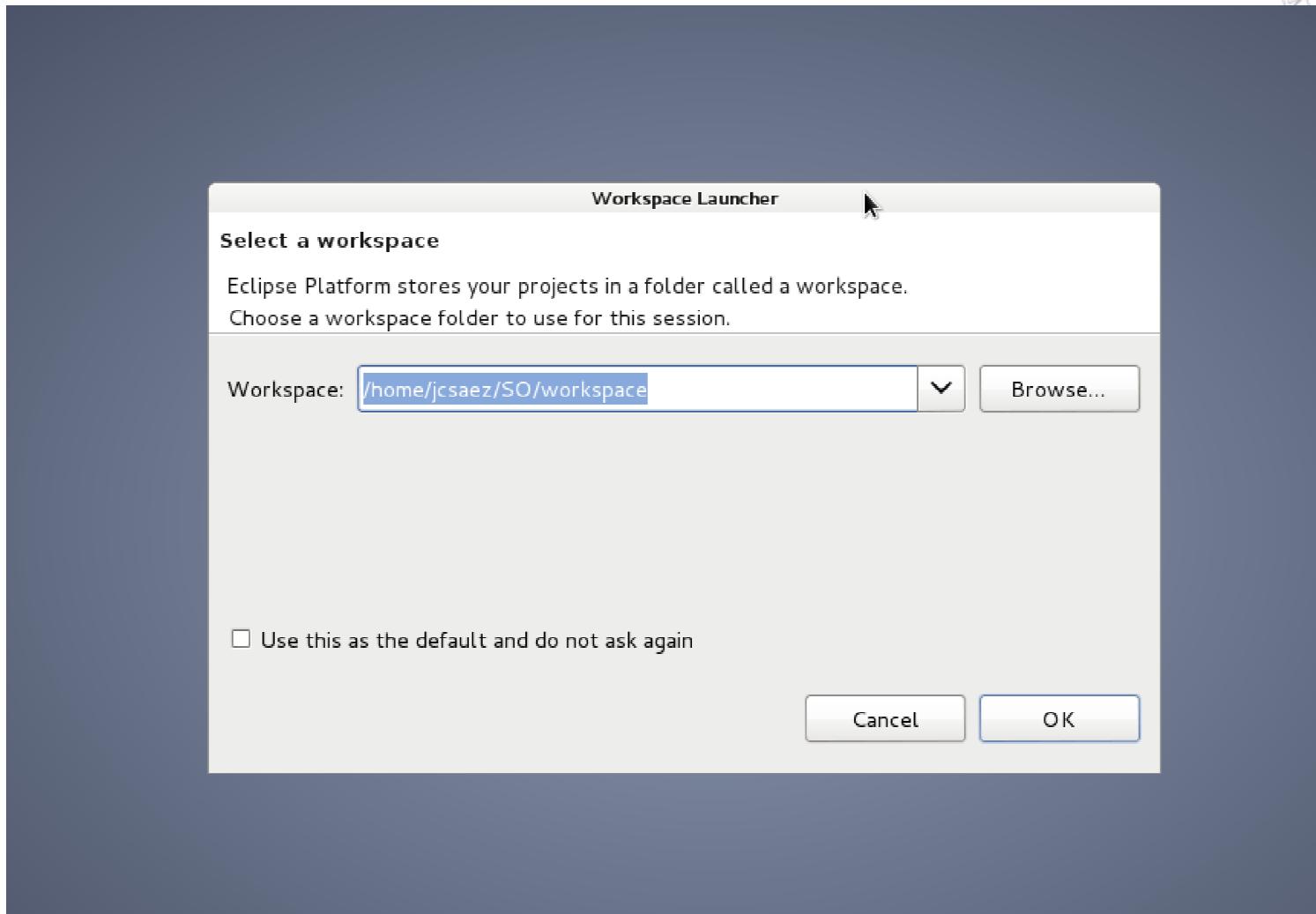
Contenido



- Alternativas para el desarrollo de aplicaciones C en Linux
- Creación de proyecto vacío + Importación
- Generación de perfil de ejecución
- Generación de perfil de depuración
- Uso de GNU make para compilar desde el terminal

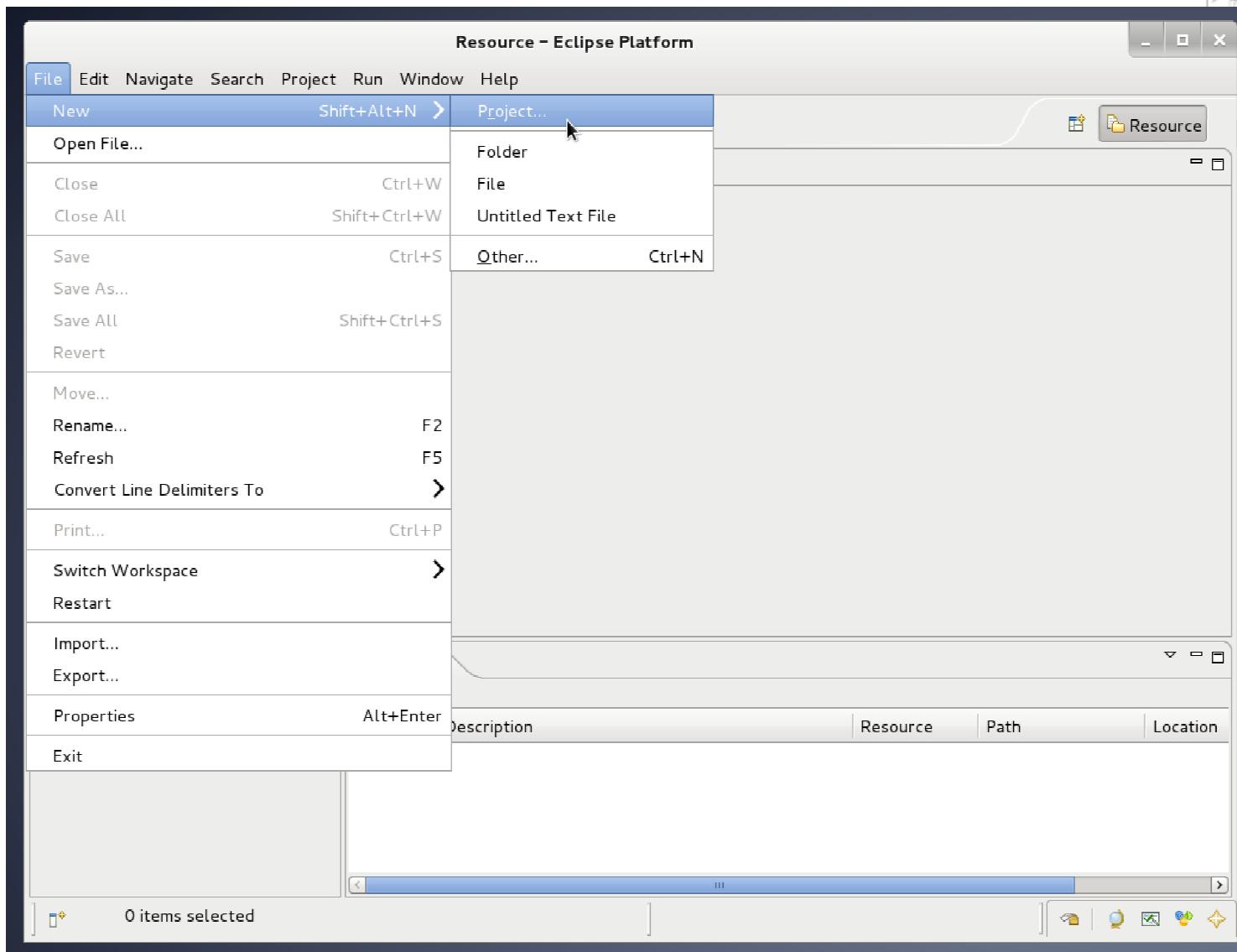


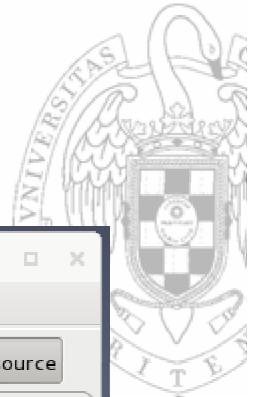
Apertura/Creación de workspace



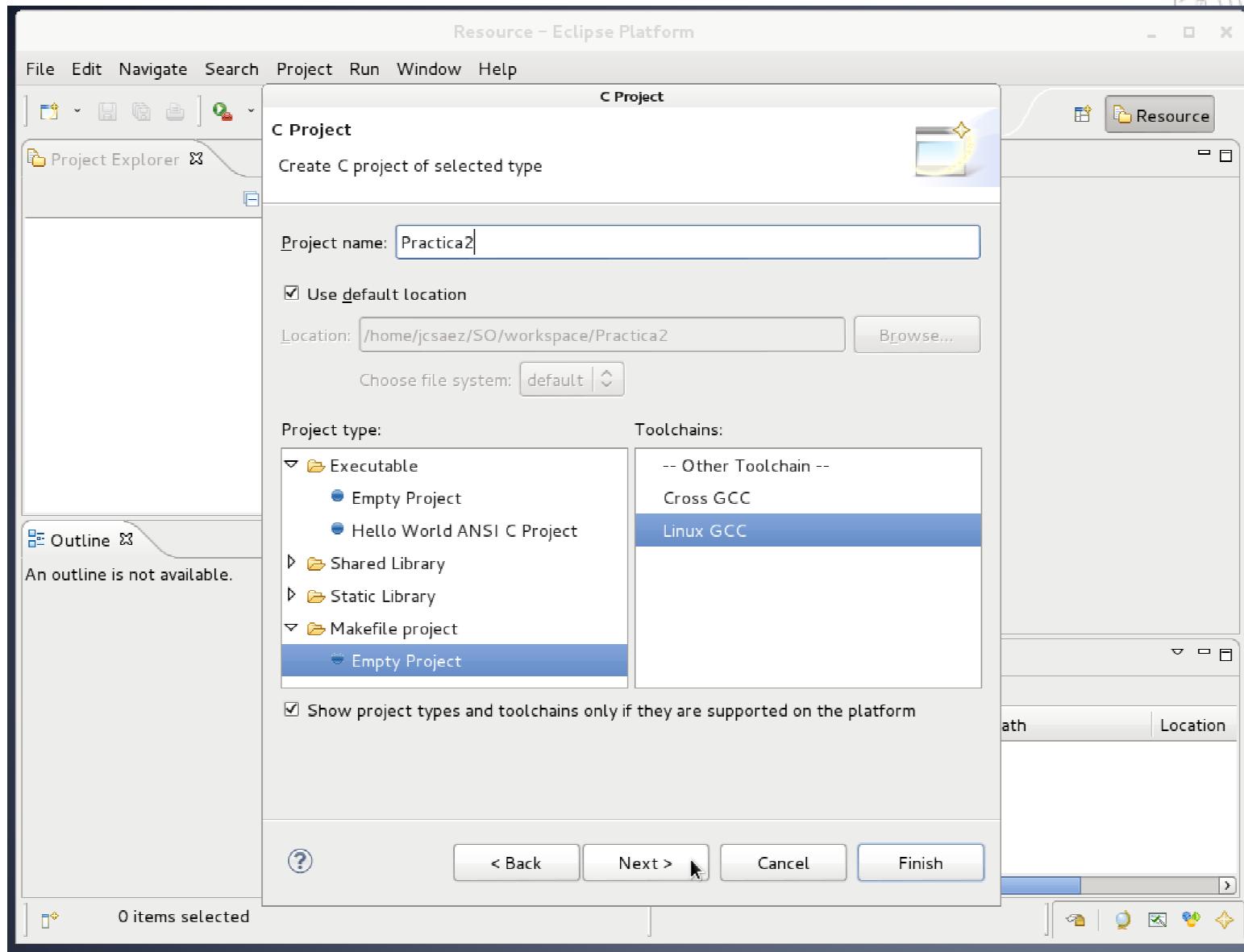


Crear proyecto vacío (I)



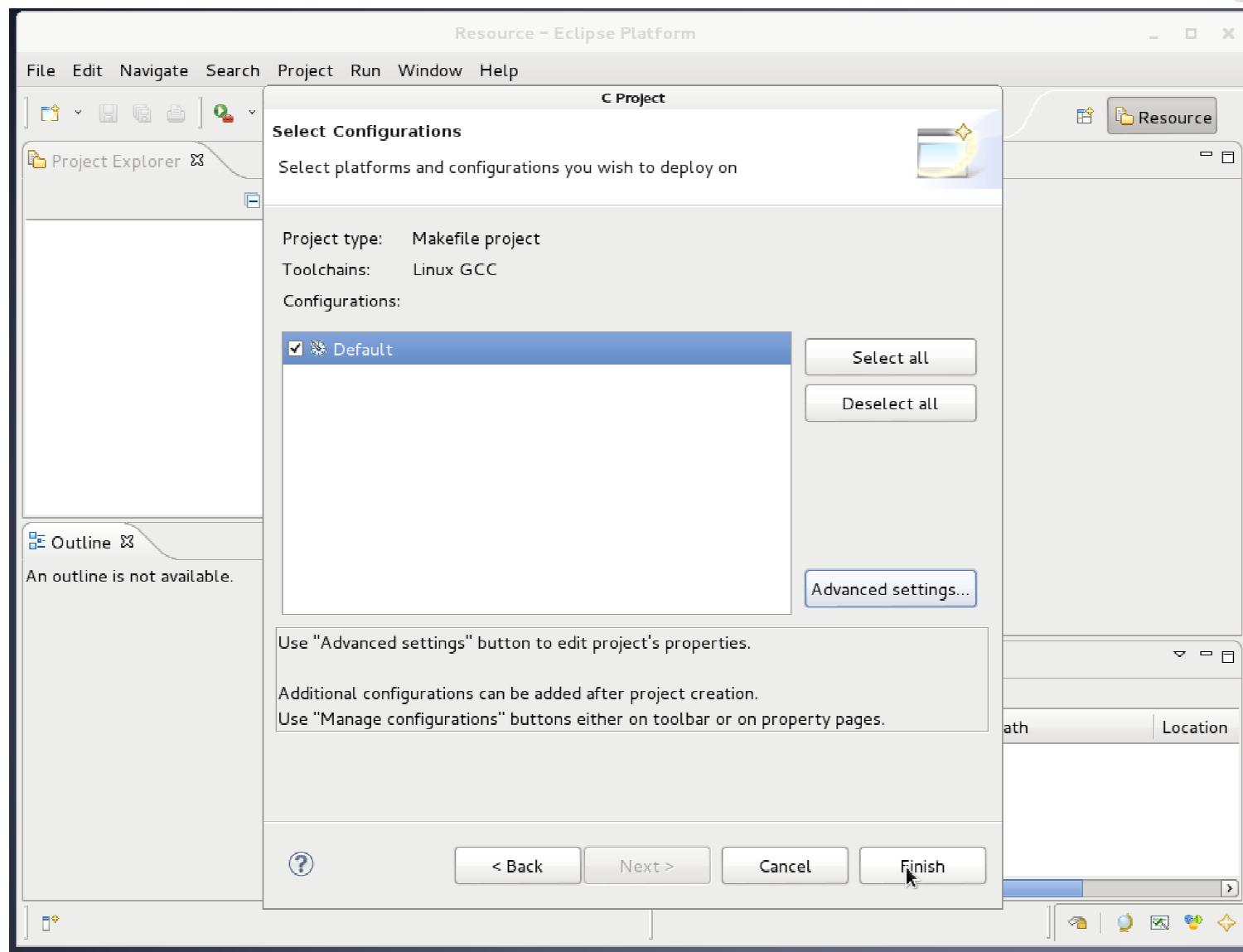


Crear proyecto vacío (II)

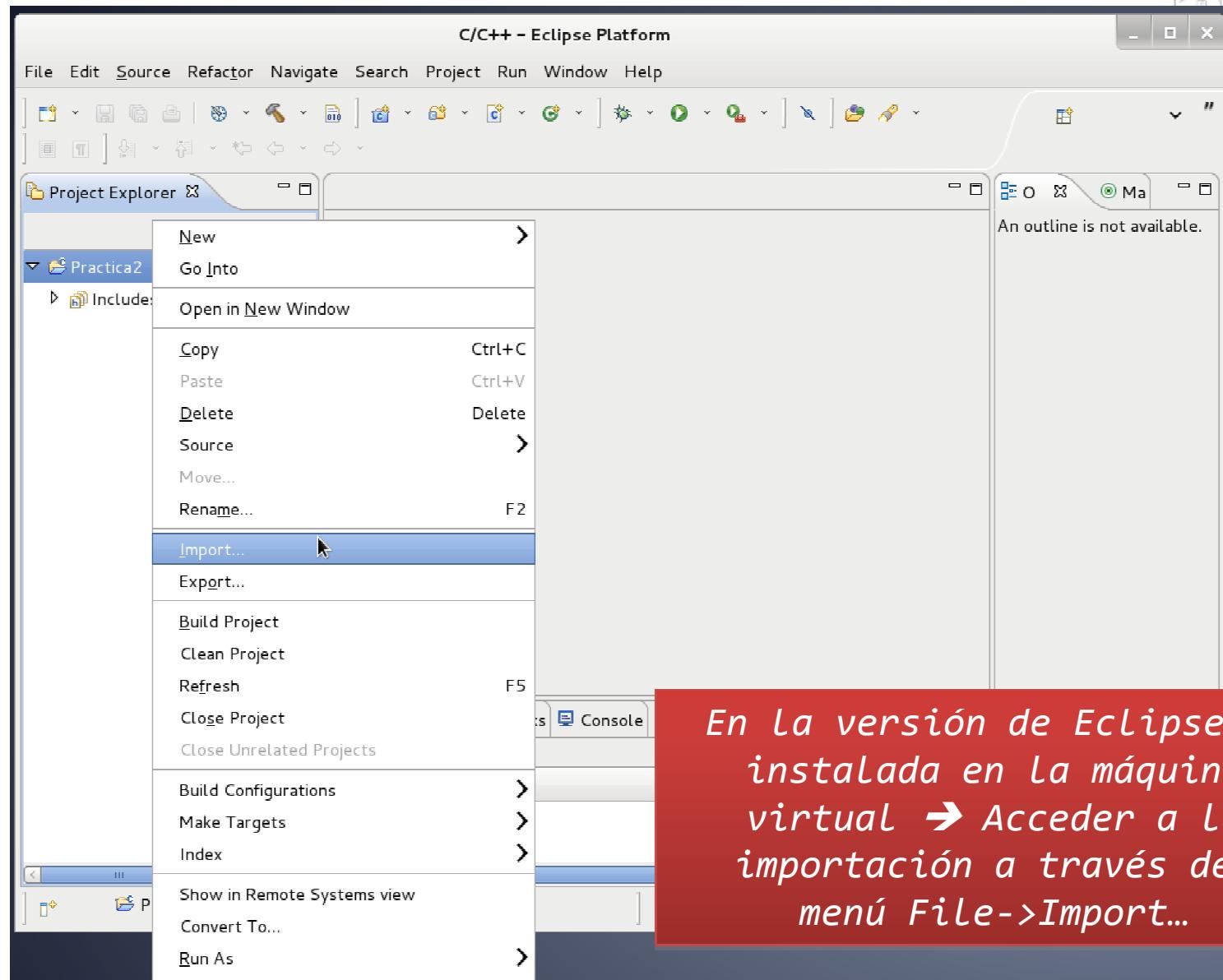




Crear proyecto vacío (III)

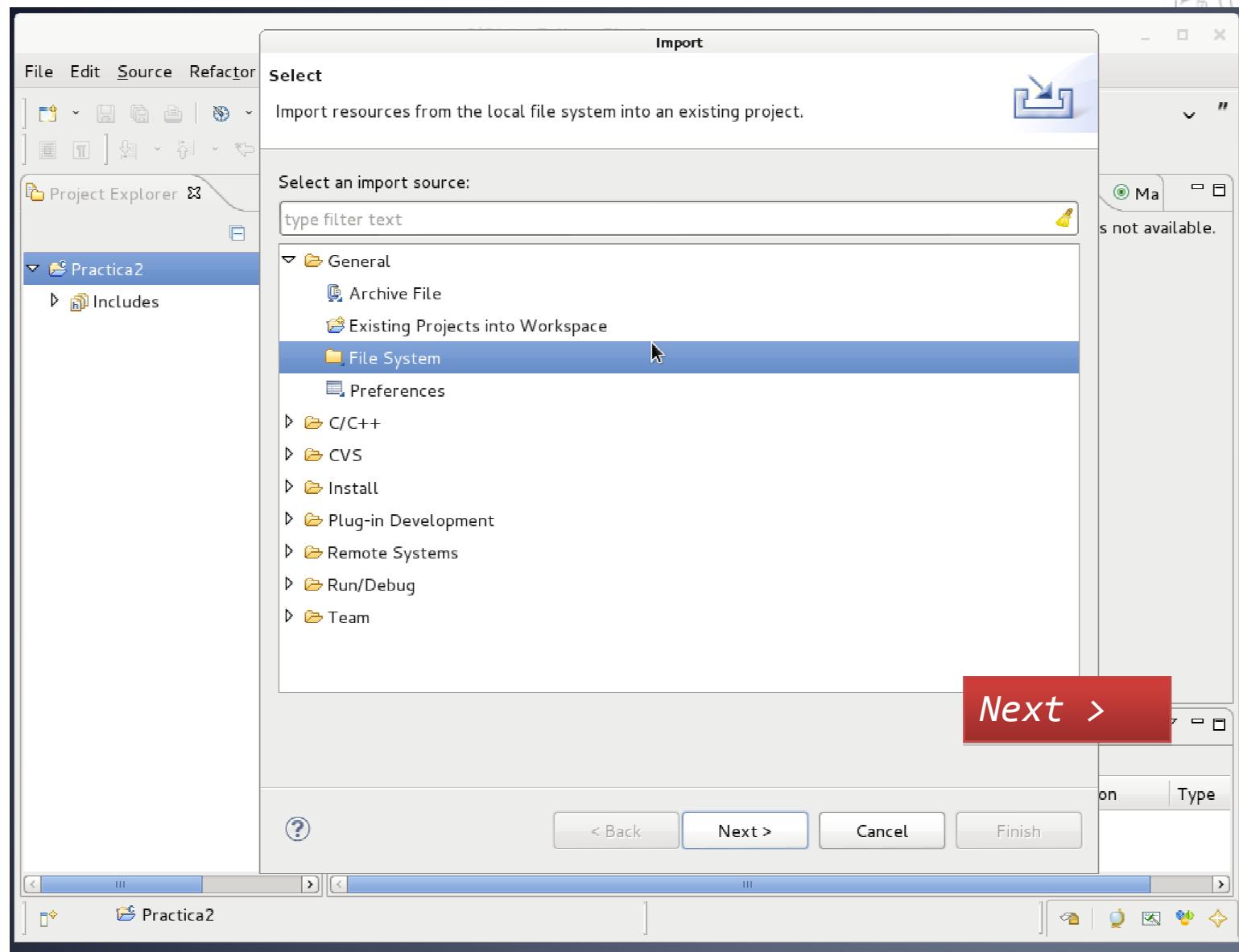


Importar proyecto con *Makefile* (I)



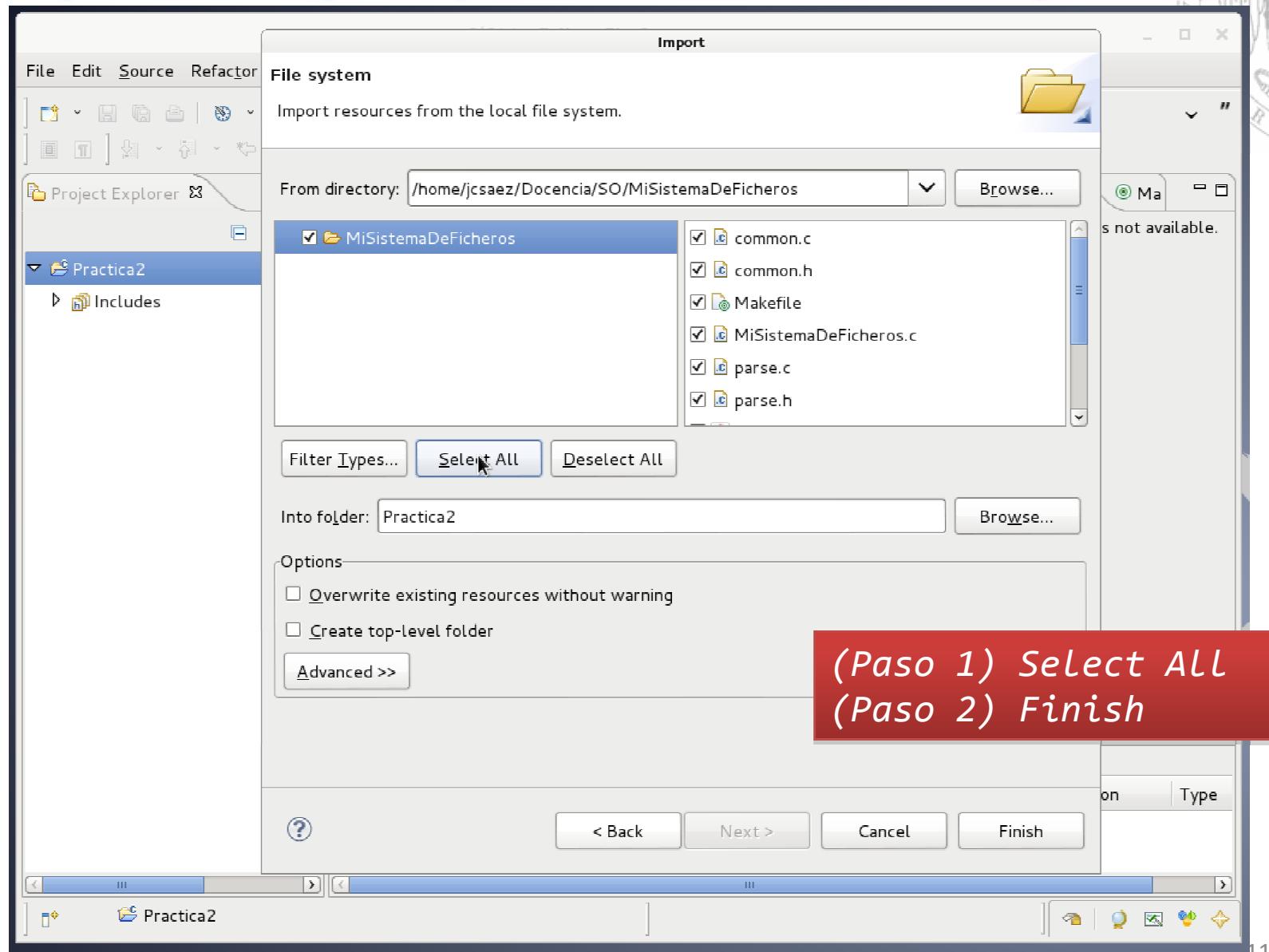
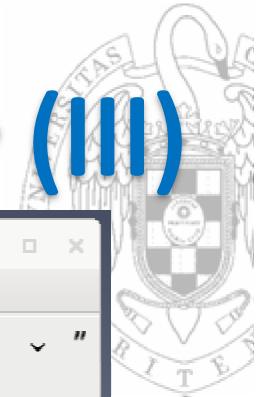


Importar proyecto con Makefile (II)





Importar proyecto con Makefile (III)

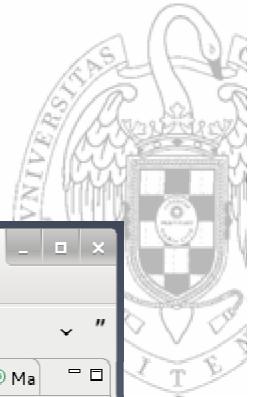




Contenido



- Alternativas para el desarrollo de aplicaciones C en Linux
- Creación de proyecto vacío + Importación
- Generación de perfil de ejecución
- Generación de perfil de depuración
- Uso de GNU make para compilar desde el terminal



Crear perfil de ejecución (I)

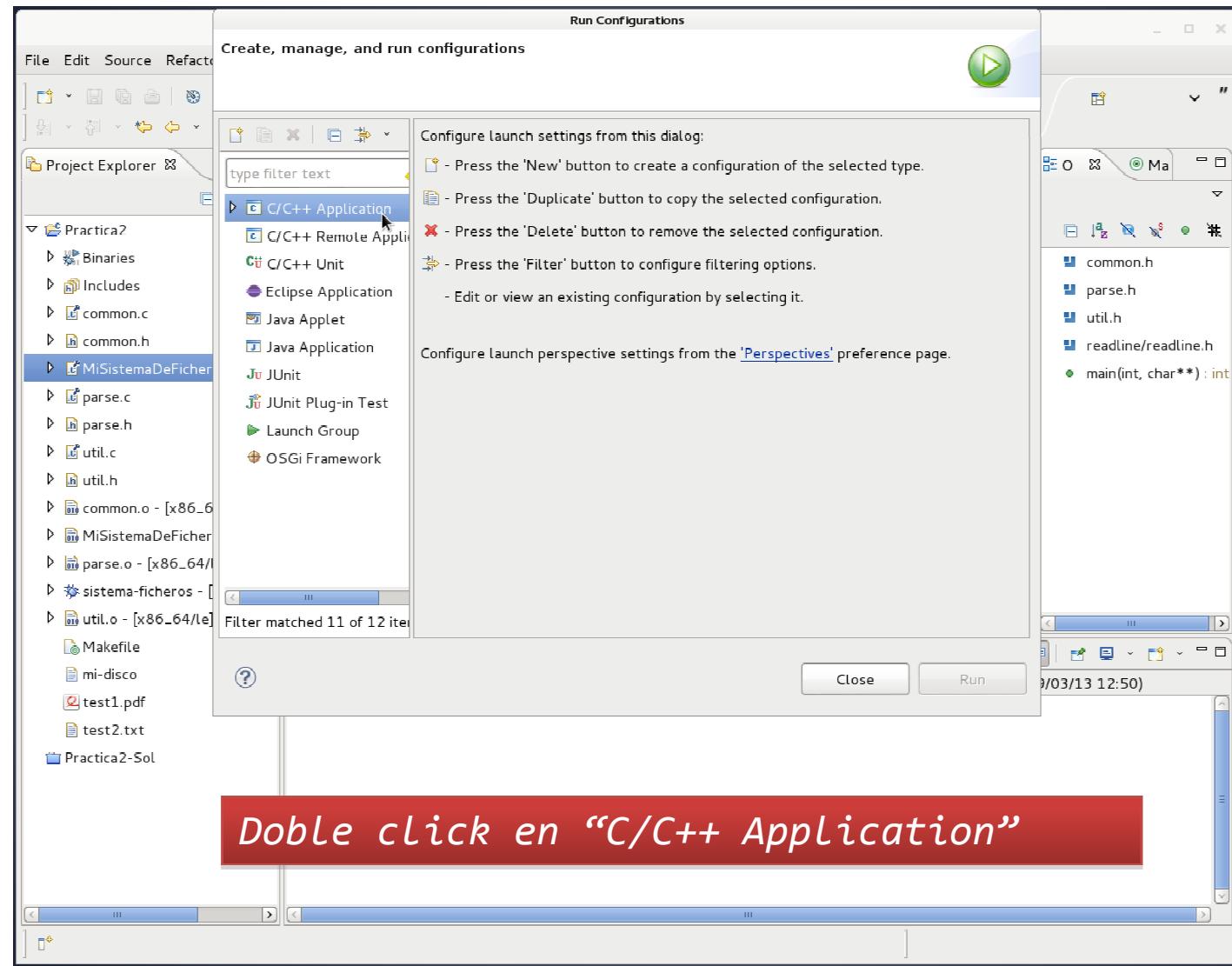
Screenshot of the Eclipse C/C++ Platform interface showing the "Run" menu open. The menu includes options like Run, Debug, Run History, Run As, Run Configurations..., and External Tools. The "Run Configurations..." option is highlighted.

The Project Explorer view shows a project named "Practica2" containing files like common.c, common.h, parse.c, parse.h, util.c, util.h, and several object files (common.o, MiSistemaDeFicheros.o, parse.o, sistema-ficheros.o, util.o). A Makefile is also present.

The Problems view shows the message: <terminated> Practica2 Default [C/C++ Application] /home/jcsaez/SO/workspace/Practica2/sistema-ficheros (08/03/13 21:41).

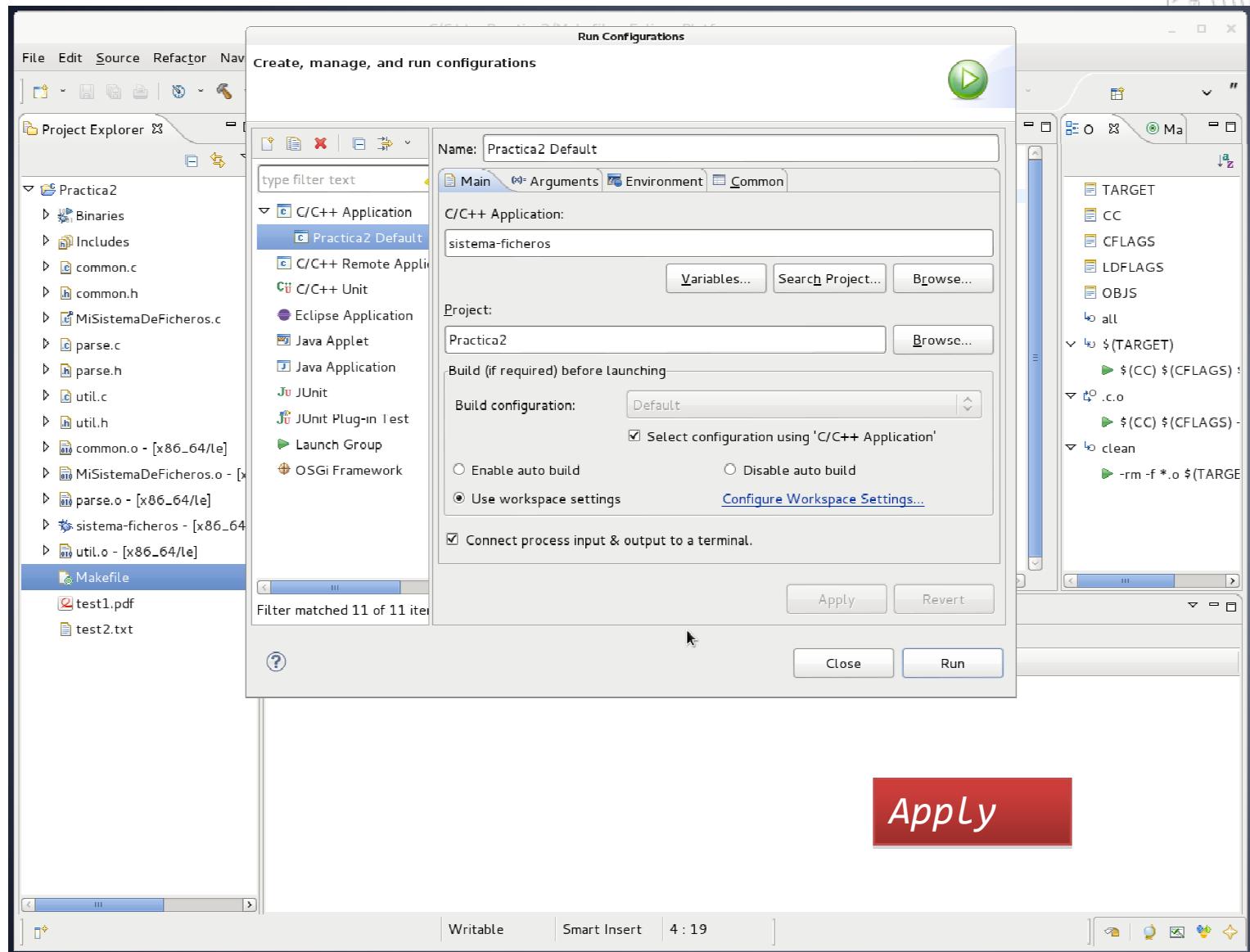


Crear perfil de ejecución (II)



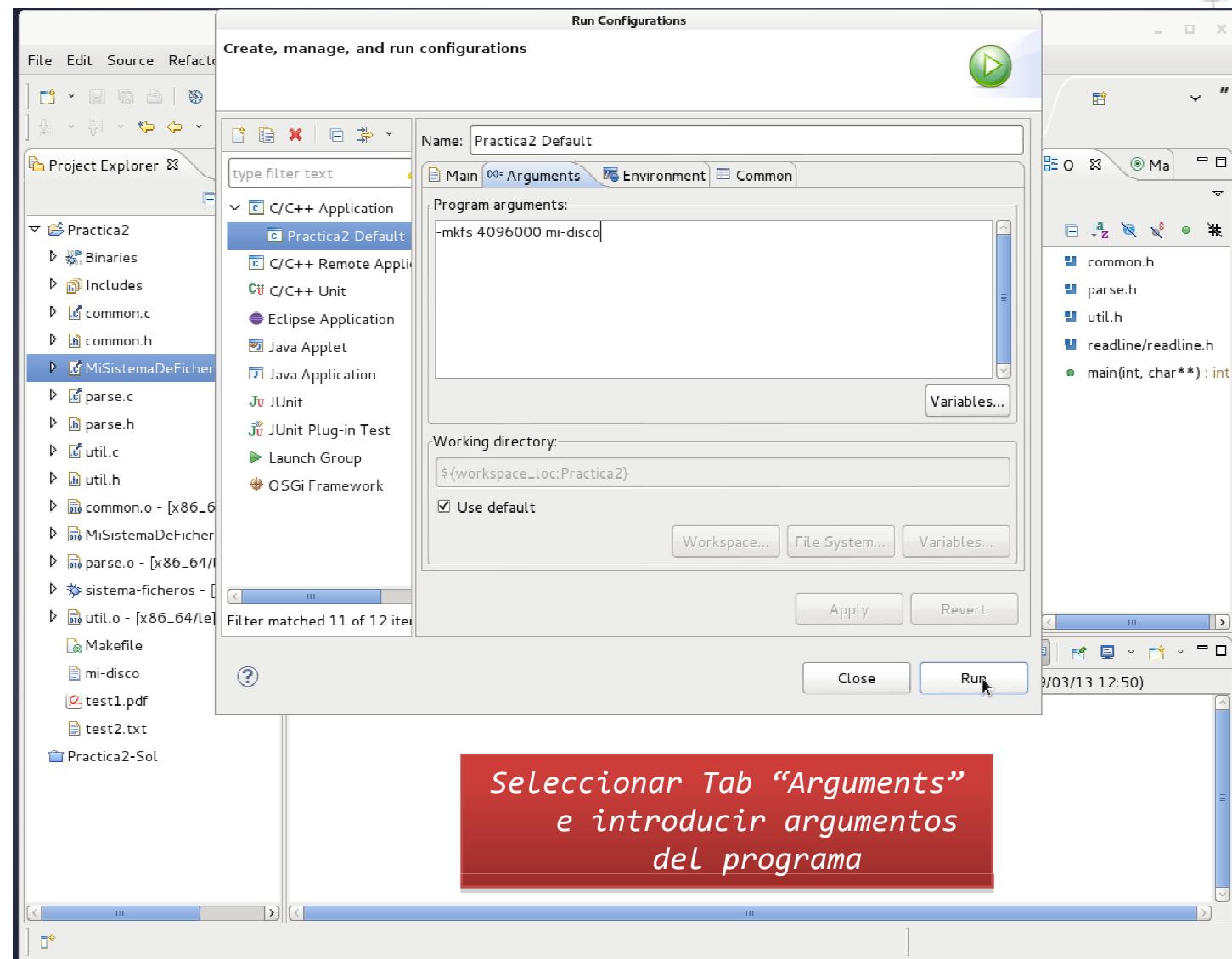


Crear perfil de ejecución (III)



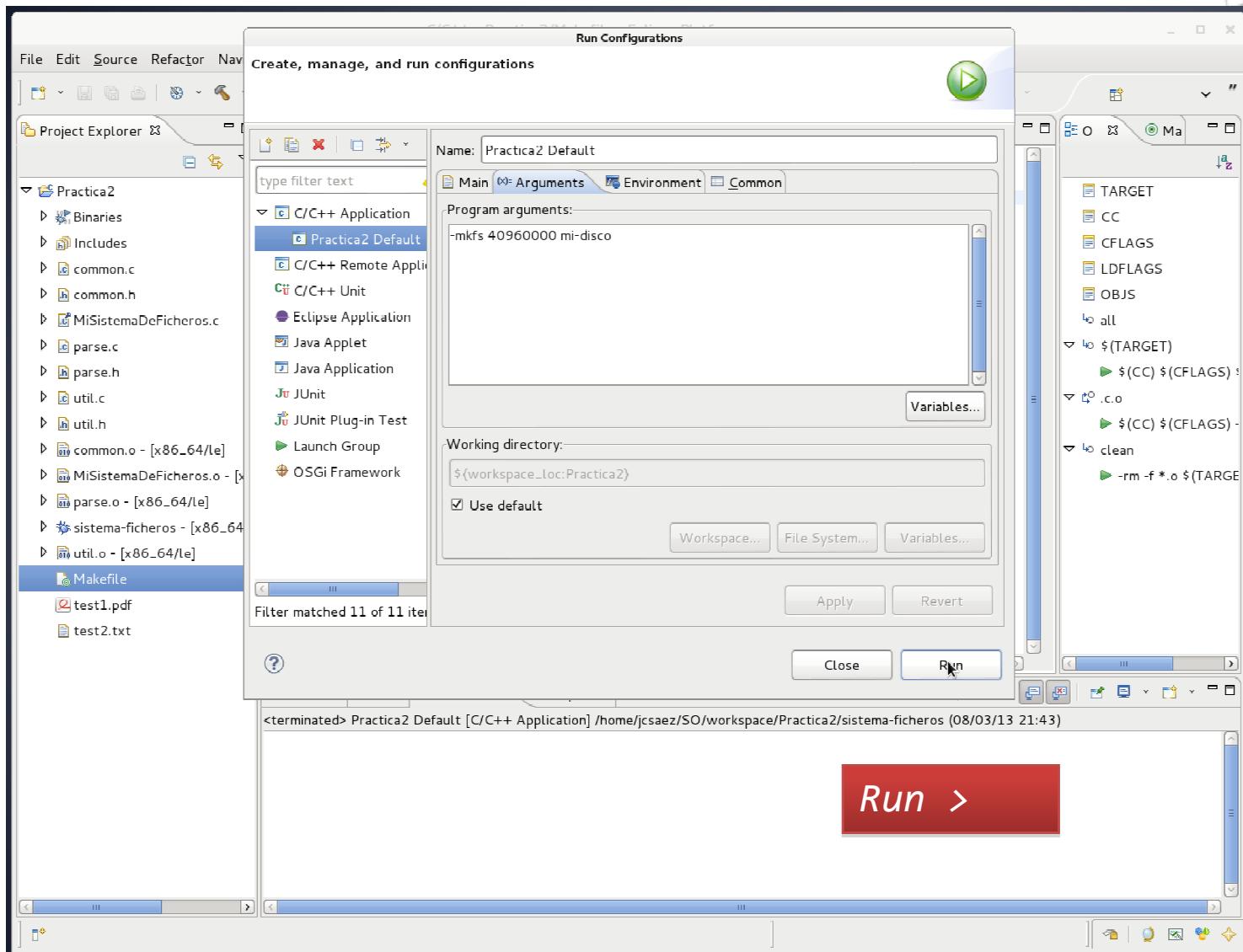


Crear perfil de ejecución (IV)



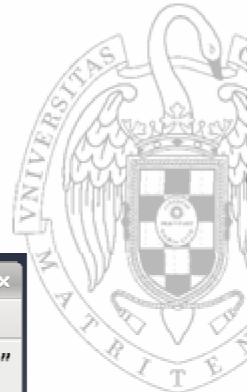


Crear perfil de ejecución (V)





Crear perfil de ejecución (II)



C/C++ - Practica2/MiSistemaDeFicheros.c - Eclipse Platform

File Edit Source Refactor Navigate Search Run Project Window Help

Project Explorer MiSistemaDeFicheros.c

```
1 #include "common.h"
2 #include "parse.h"
3 #include "util.h"
4 #include <readline/readline.h>
5
6 int main(int argc, char** argv) {
7     MiSistemaDeFicheros miSistemaDeFicheros;
8     miSistemaDeFicheros.numNodosLibres = MAX_NODOSI;
9
10    char* lineaComando;
11    parseInfo* info; // Almacena toda la información que retorna el parser
12    struct commandType* comando; // Almacena el comando y la lista de argumentos
13    int ret; // Código de retorno de las llamadas a funciones
14
15    if ((argc == 4) && (strcmp(argv[1], "-mkfs") == 0)) {
16        // ./MiSistemaDeFicheros -mkfs tamDisco nombreArchivo
17        ret = myMkfs(&miSistemaDeFicheros, atoi(argv[2]), argv[3]);
18        if (ret) {
19            fprintf(stderr, "Incapaz de formatear, código de error: %d\n", ret);
20            exit(-1);
21        }
22    } else {
23        fprintf(stderr, "Error, debes introducir el tamaño del disco y su nombre:");
24        exit(-1);
25    }
26    initNodosI(&miSistemaDeFicheros);
27    fprintf(stderr, "Sistema de ficheros disponible\n");
28}
```

util.c MiSistemaDeFicheros.c

common.h parse.h util.h readline/readline.h

main(int, char**): int

Problems Tasks Console Properties

<terminated> Practica2 Default [C/C++ Application] /home/jcsaez/SO/workspace/Practica2/sistema-ficheros (09/03/13 12:48)
sistema-ficheros: util.c:65: myMkfs: Assertion `myQuota(miSistemaDeFicheros) >= 1' failed.



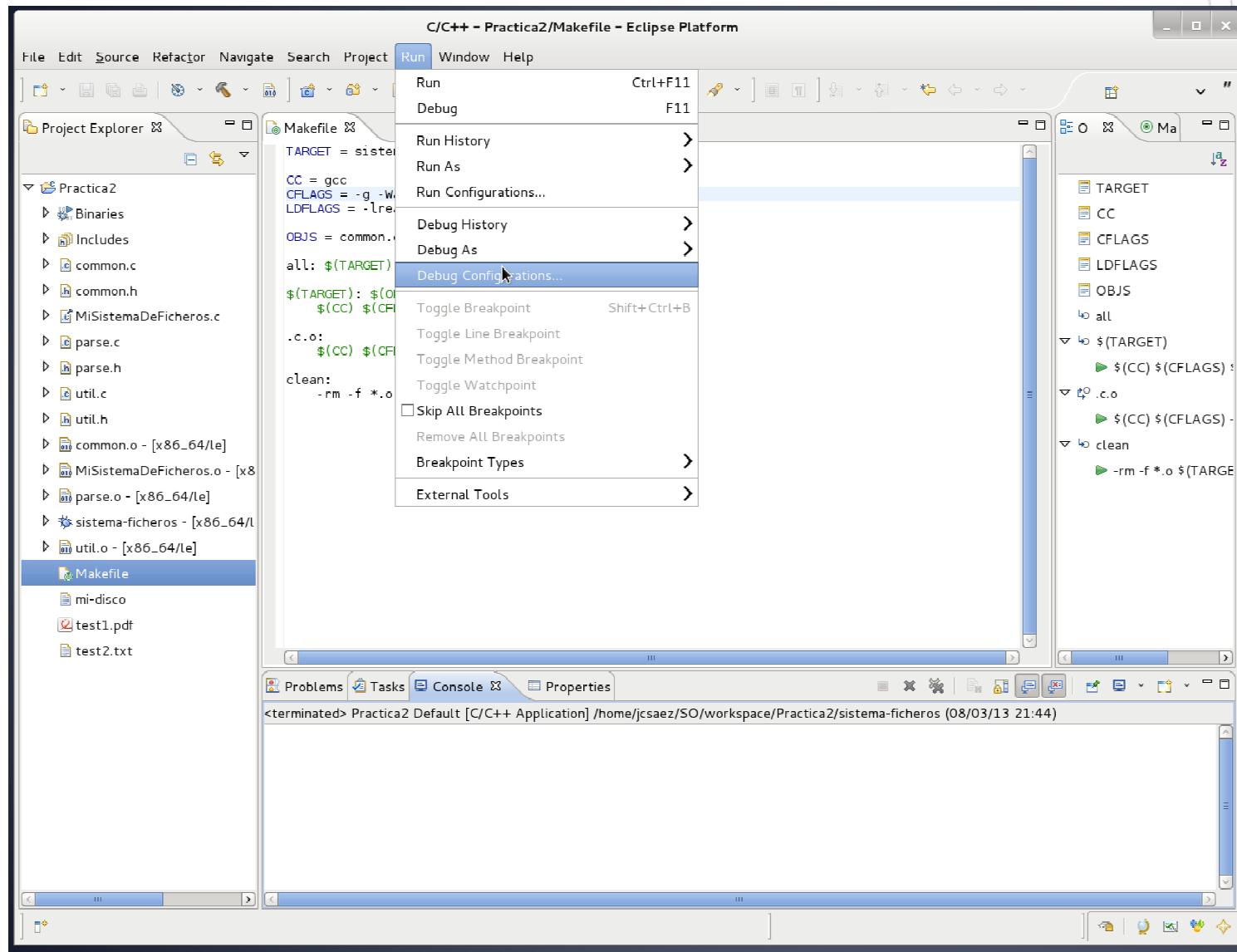
Contenido



- Alternativas para el desarrollo de aplicaciones C en Linux
- Creación de proyecto vacío + Importación
- Generación de perfil de ejecución
- Generación de perfil de depuración
- Uso de GNU make para compilar desde el terminal

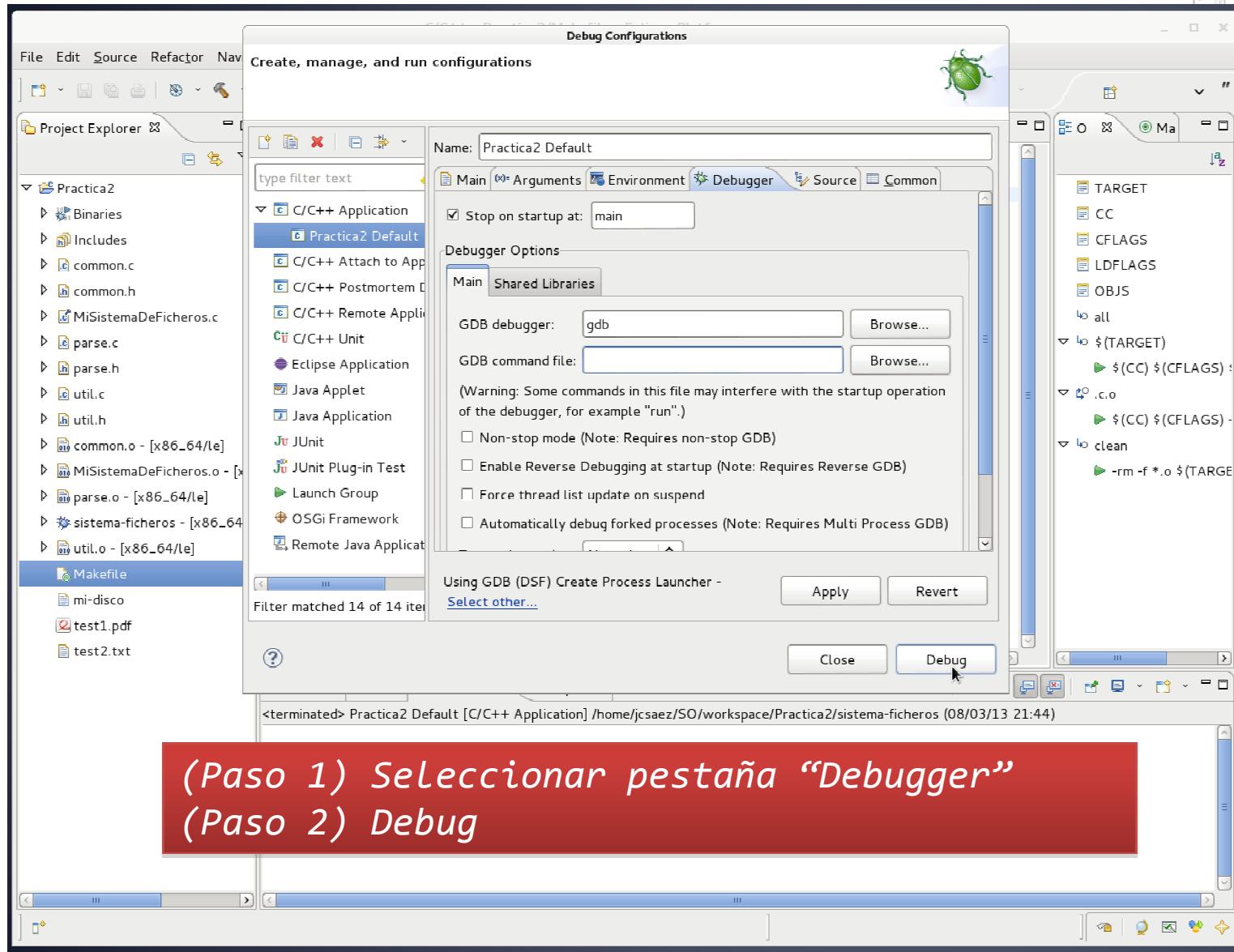


Crear perfil de depuración (I)



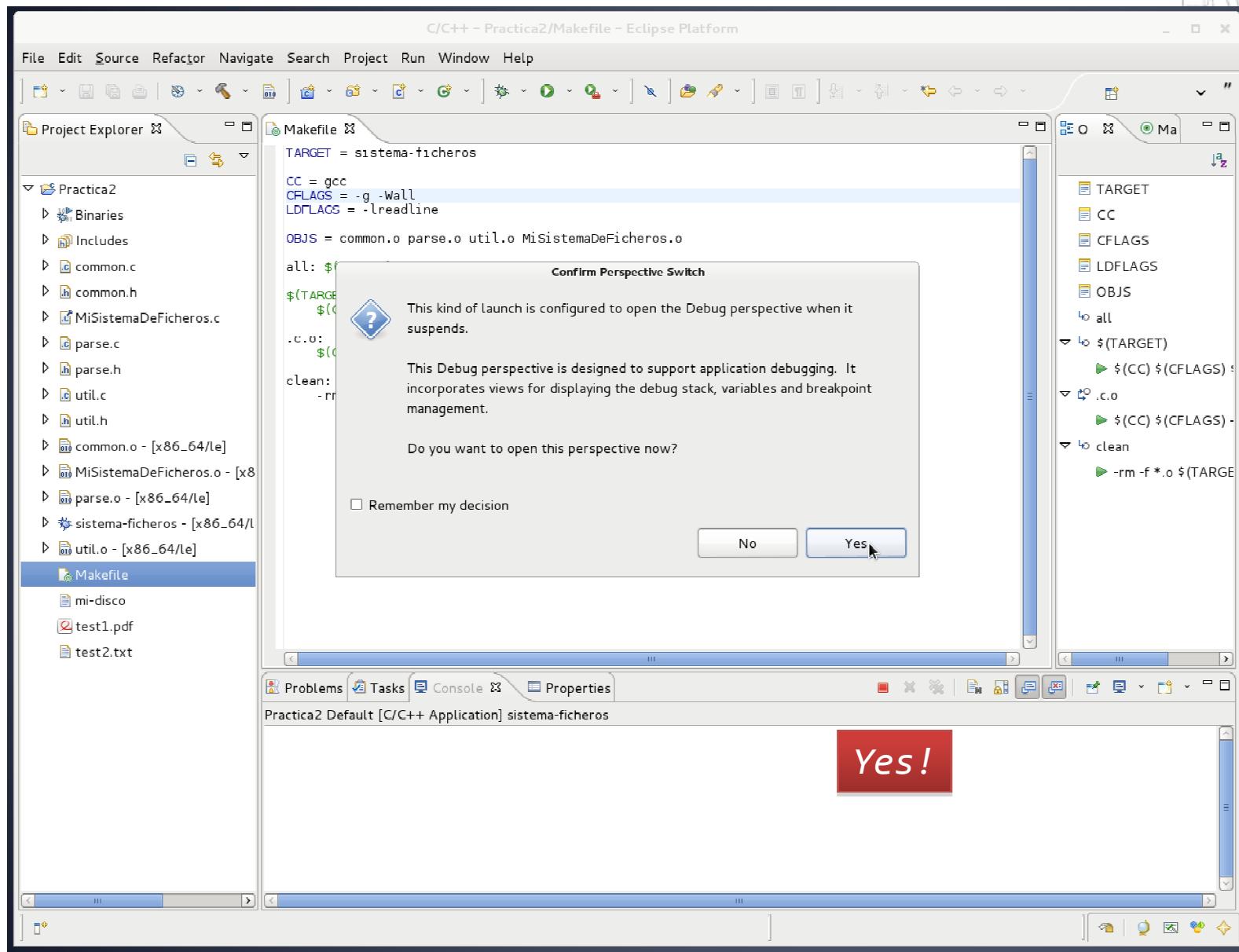
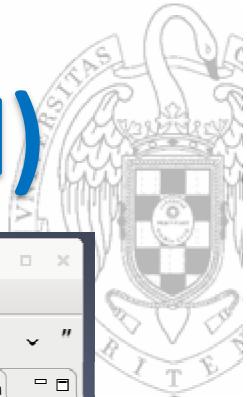


Crear perfil de depuración (II)





Crear perfil de depuración (III)



Crear perfil de depuración (IV)



Screenshot of the Eclipse Platform interface showing the Debug perspective. The left pane displays the stack trace: 'Practica2 Default [C/C++ Application] / sistema-ficheros [8246] [cores: 0] / Thread [1] 8246 [core: 0] (Suspended : Breakpoint) main() at MiSistemaDeFicheros.c:8 0x4023bd'. The bottom-left pane shows the source code of 'MiSistemaDeFicheros.c' with a breakpoint at line 10. A red arrow points to the 'gdb' tab in the bottom-left toolbar, which is highlighted. The bottom-right pane shows the 'Console' tab with a list of gdb commands.

```
#include "common.h"
#include "parse.h"
#include "util.h"
#include <readline/readline.h>

int main(int argc, char** argv) {
    MiSistemaDeFicheros miSistemaDeFicheros;
    miSistemaDeFicheros.numNodosLibres = MAX_NODOSI;

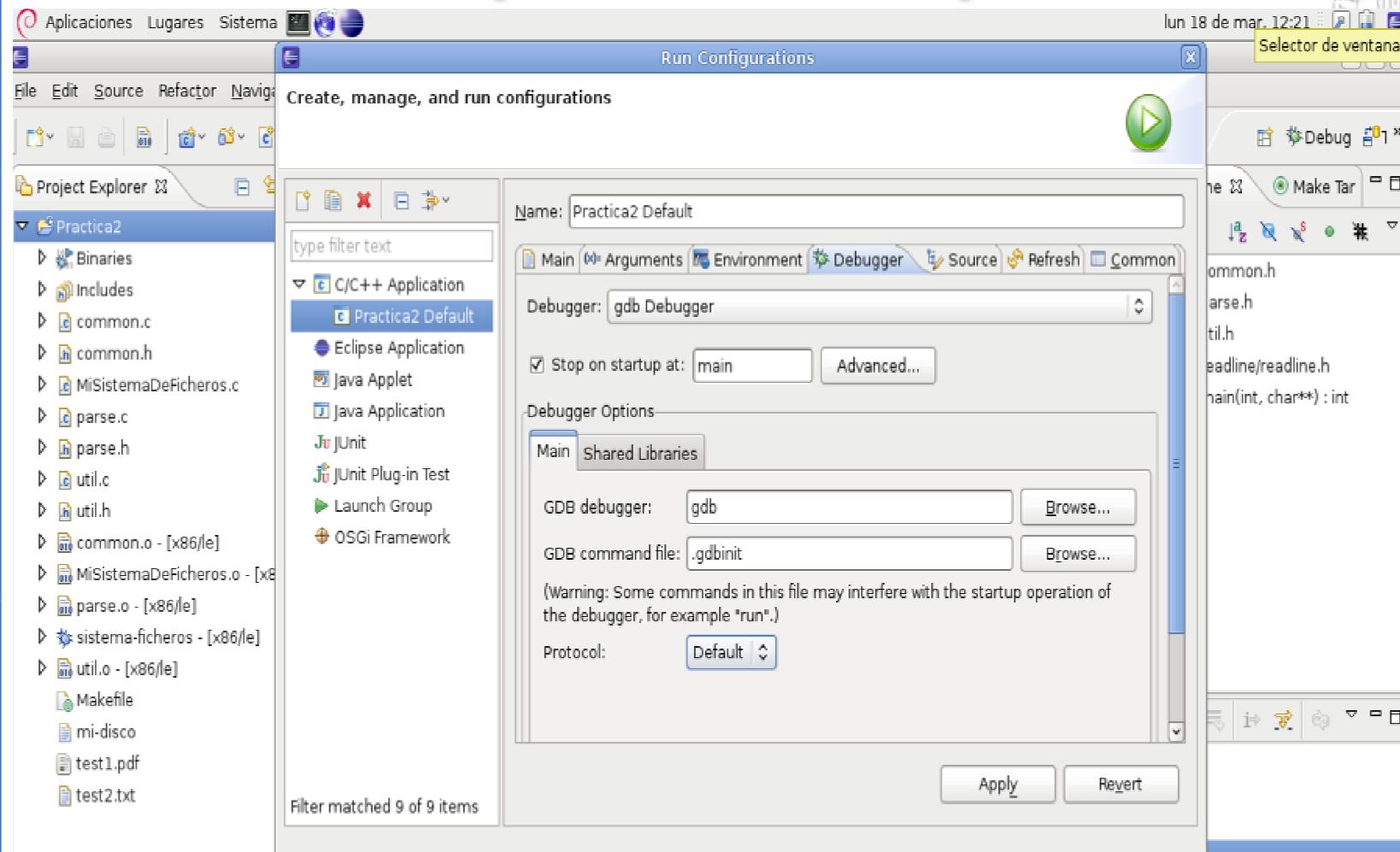
    char* lineaComando;
    parseInfo* info; // Almacena toda la información que retorna el parser
    struct commandType* comando; // Almacena el comando y la lista de argumentos

    if ((argc == 4) && (argv[1][1] == 'm')) { // Por qué comprobar todo el argumento :-)?
        // ./MiSistemaDeFicheros .mkfs tamDisco nombreArchivo
        if (!myMkfs(&miSistemaDeFicheros, atoi(argv[2]), argv[3])) {
            fprintf(stderr, "Incapaz de formatear\n");
            exit(-1);
        }
    }
}
```

Es posible introducir comandos de gdb desde la consola seleccionando el modo “gdb”



MV: Crear perfil de depuración (II)



(Paso 1) Seleccionar pestaña “Debugger”

(Paso 2) Seleccionar “gdb Debugger” en Lista desplegable

(Paso 3) > Apply



MV: Crear perfil de depuración (III)



Cambiar manualmente La perspectiva haciendo click en el icono y seleccionando “Debug” en el menú desplegable

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer MiSistemaDeFicheros.c

```
miSistemaDeFicheros miSistemaDeFicheros;
miSistemaDeFicheros.numNodosLibres = MAX_NODOSI;

char* lineaComando;
parseInfo* info; // Almacena toda la información que retorna el parser
struct commandType* comando; // Almacena el comando y la lista de argumentos
int ret; // Código de retorno de las llamadas a funciones

if ((argc == 4) && (strcmp(argv[1], "-mkfs") == 0)) {
    // ./MiSistemaDeFicheros -mkfs tamDisco nombreArchivo
    ret = myMkfs(&miSistemaDeFicheros, atoi(argv[2]), argv[3]);
    if (ret) {
        fprintf(stderr, "Incapaz de formatear, código de error: %d\n", ret);
        exit(-1);
    }
} else {
    fprintf(stderr, "Error, debes introducir el tamaño del disco y su nombre:");
    exit(-1);
}
```

Debug Team Synchronizing Other... common.h parse.h util.h readline/readline.h main(int, char**) : int

Output View:

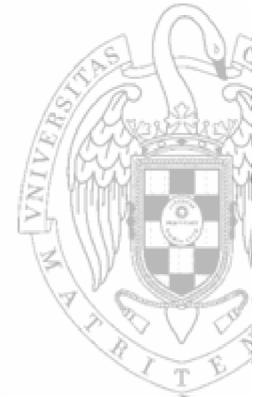
```
* <terminated>gdb Debugger (18/03/13 12:17) (Exited, exit code = 0)
* <terminated, exit value: 0>gdb (18/03/13 12:17)
* <terminated, exit value: 0>/home/alumno/workspace/Practica2/sistema-ficheros (18/03/13 12:17)
```



Contenido



- Alternativas para el desarrollo de aplicaciones C en Linux
- Creación de proyecto vacío + Importación
- Generación de perfil de ejecución
- Generación de perfil de depuración
- Uso de GNU make para compilar desde el terminal



Usando make

```
jcsaez@debian64:~/SO/workspace/Practica2$ make
gcc -g -Wall -I. -c common.c
gcc -g -Wall -I. -c parse.c
gcc -g -Wall -I. -c util.c
util.c: In function 'myMkfs':
util.c:12:18: warning: variable 'nodoActual' set but not used [-Wunused-but-set-variable]
util.c: In function 'myExport':
util.c:133:14: warning: 'handle' is used uninitialized in this function
gcc -g -Wall -I. -c MiSistemaDeFicheros.c
gcc -g -Wall -lreadline -o sistema-ficheros common.o parse.o util.o MiSistemaDeFicheros.o
jcsaez@debian64:~/SO/workspace/Practica2$ ./sistema-ficheros -mkfs 4096000 mi-disco
sistema-ficheros: util.c:65: myMkfs: Assertion `myQuota(miSistemaDeFicheros) >= 1' failed.
Abortado
jcsaez@debian64:~/SO/workspace/Practica2$ make clean
rm -f *.o sistema-ficheros
jcsaez@debian64:~/SO/workspace/Practica2$
```

Compilar+Enlazar programa

Ejecutar el programa

Borrar ficheros resultantes de la compilación