



# Lab introduction

## Introduction to computers II

**Daniel Báscones García**

**José Manuel Mendías Cuadros**

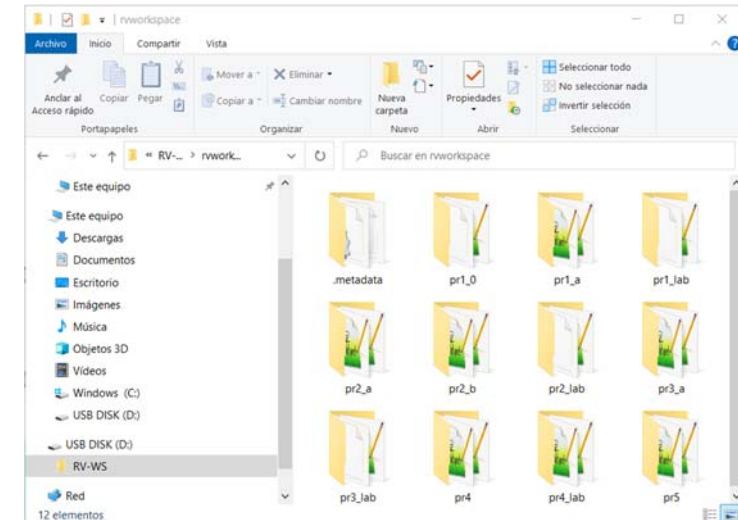
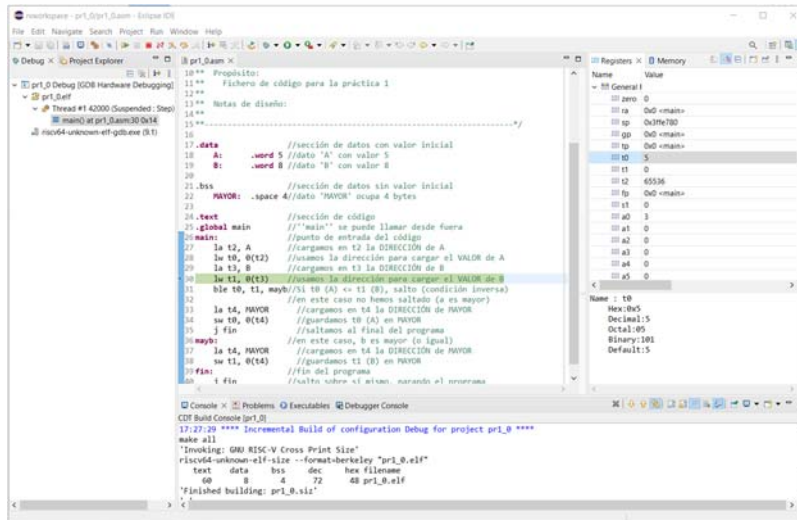
*Dpto. Arquitectura de Computadores y Automática  
Universidad Complutense de Madrid*



# Material



15/01/23 version



Lab introduction

FC-2

2


Eclipse IDE  
for Embedded C/C++ Developers

Workspace

# Installing the environment at home



## IDE installation

- Download the EclipseRV.zip file to C:/
- Unzip this file
- Go to the C:/EclipseRV/eclipse folder
- Create a direct access to the  **eclipse** file, and name it EclipseRV.
- Move this direct access to a convenient location
- Delete the EclipseRV.zip file

## Workspace installation

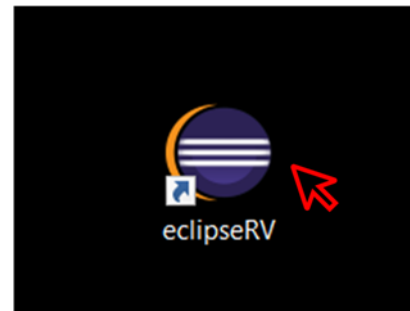
- Connect a pendrive
- Download the FC2labWS.zip to D:/
- Unzip this file
- Delete the FC2labWS.zip file



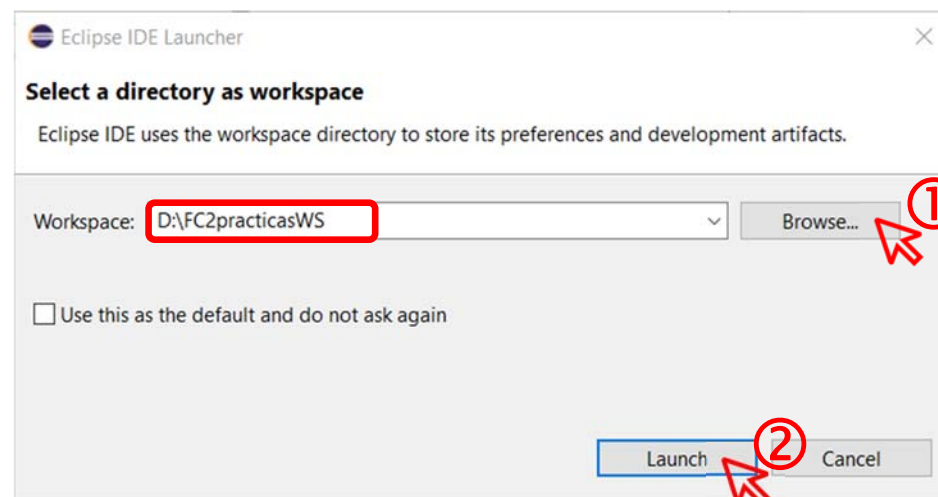
# IDE start up

## At home

- Click on the EclipseRV [direct access](#) created during the installation.



- In the dialog box, [select the downloaded Workspace](#).

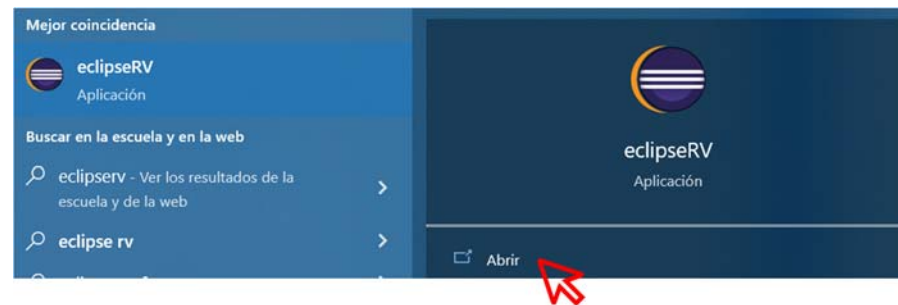


# IDE start up

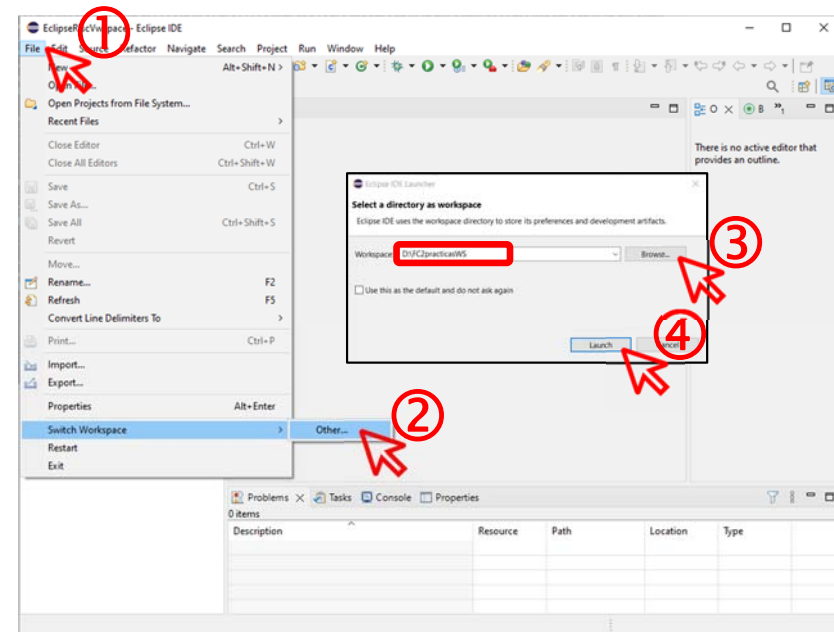
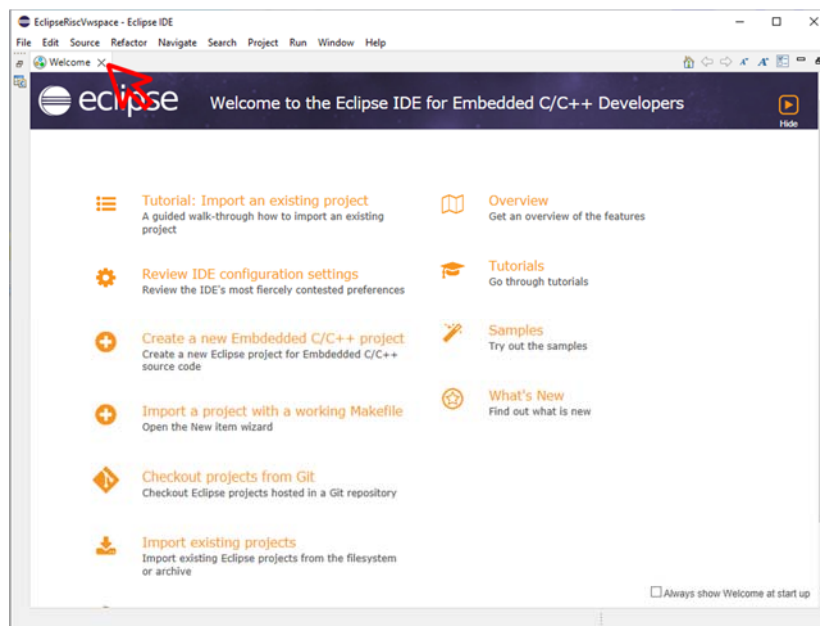
## In the lab



- Execute EclipseRV from the **Windows browser**.

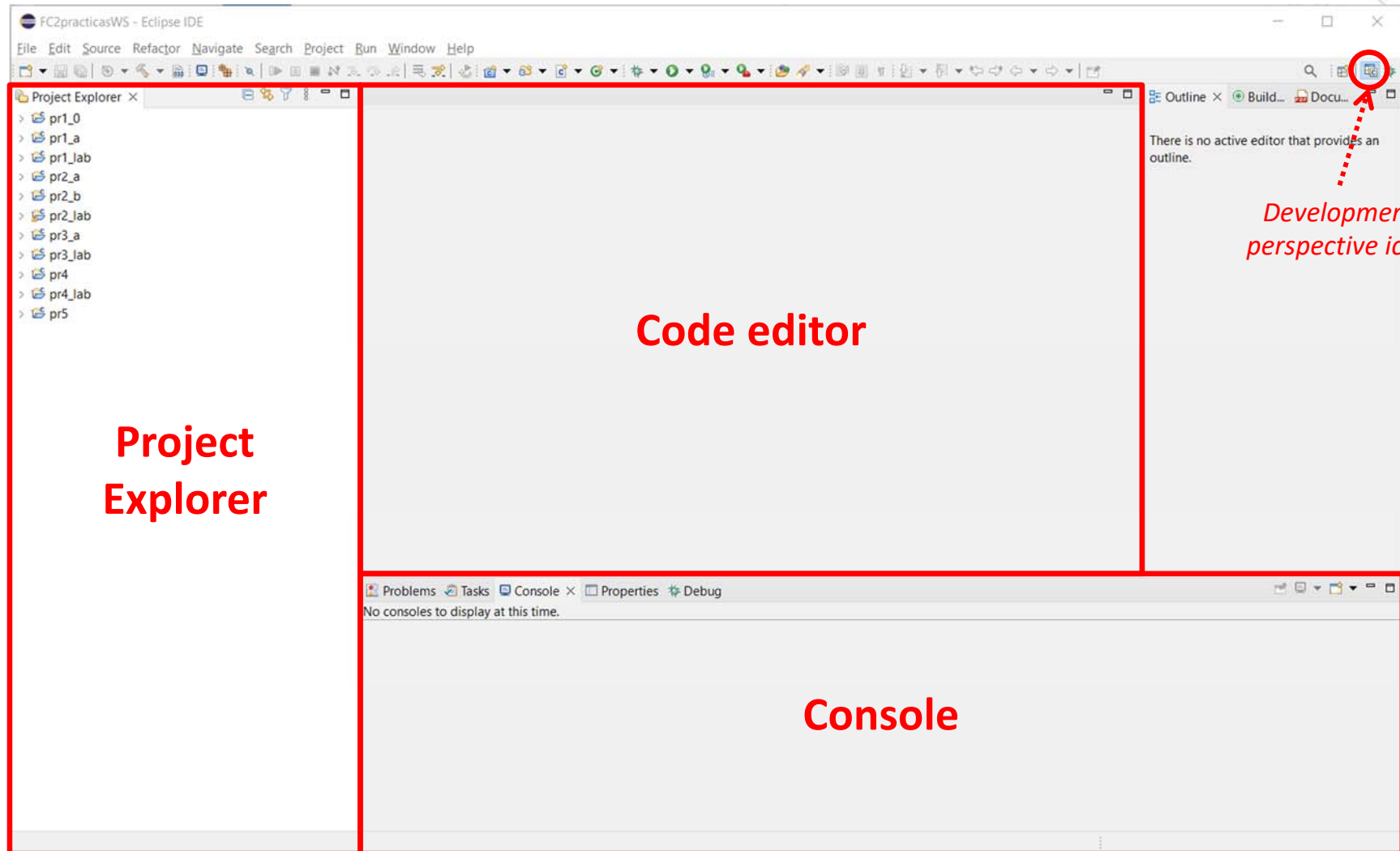


- Close the presentation window and **switch to the downloaded Workspace**.



# Development perspective

## Main views



**Project Explorer**

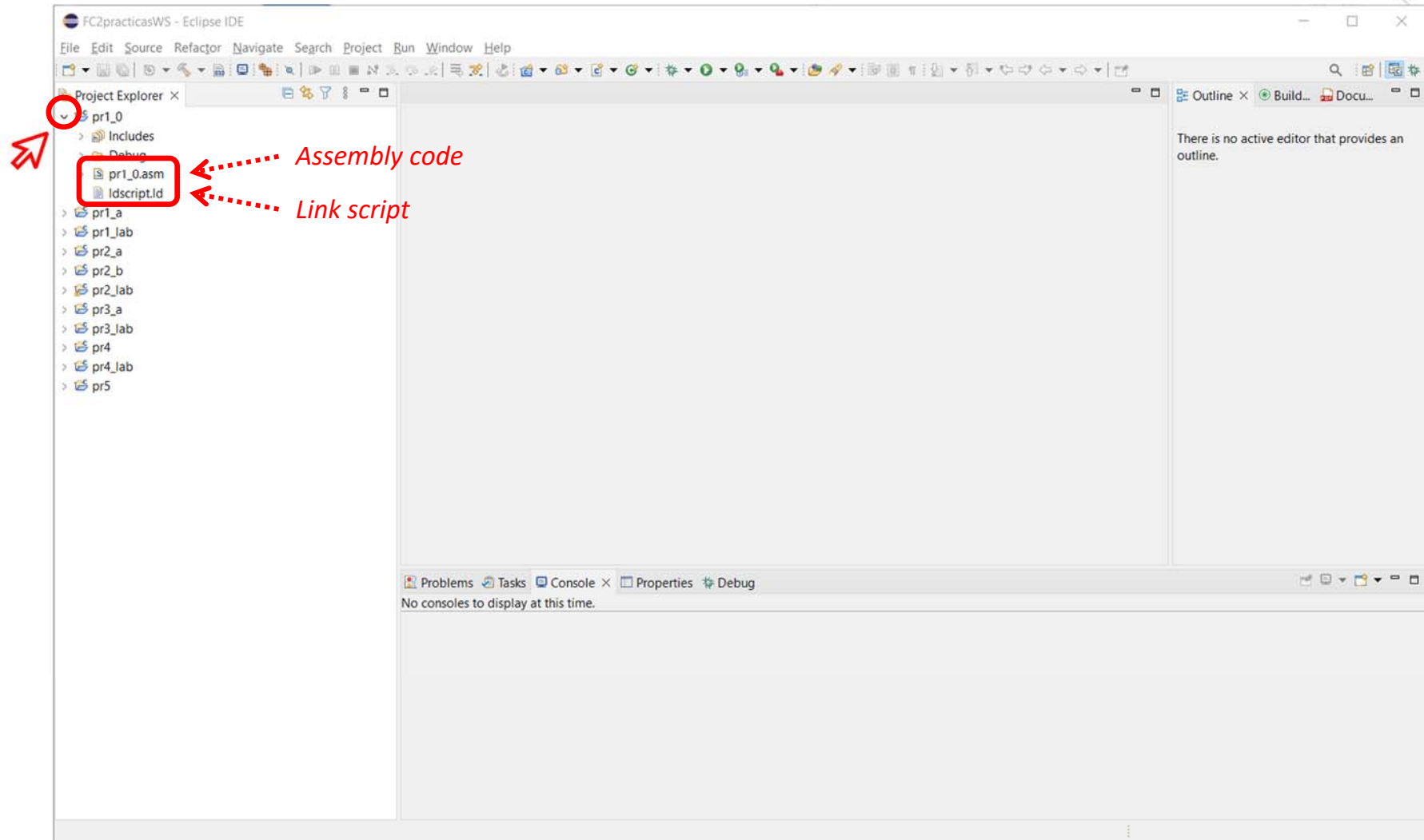
**Code editor**

**Console**

*Development perspective icon*

# Development perspective

## Project files





# Development perspective

## Code editor



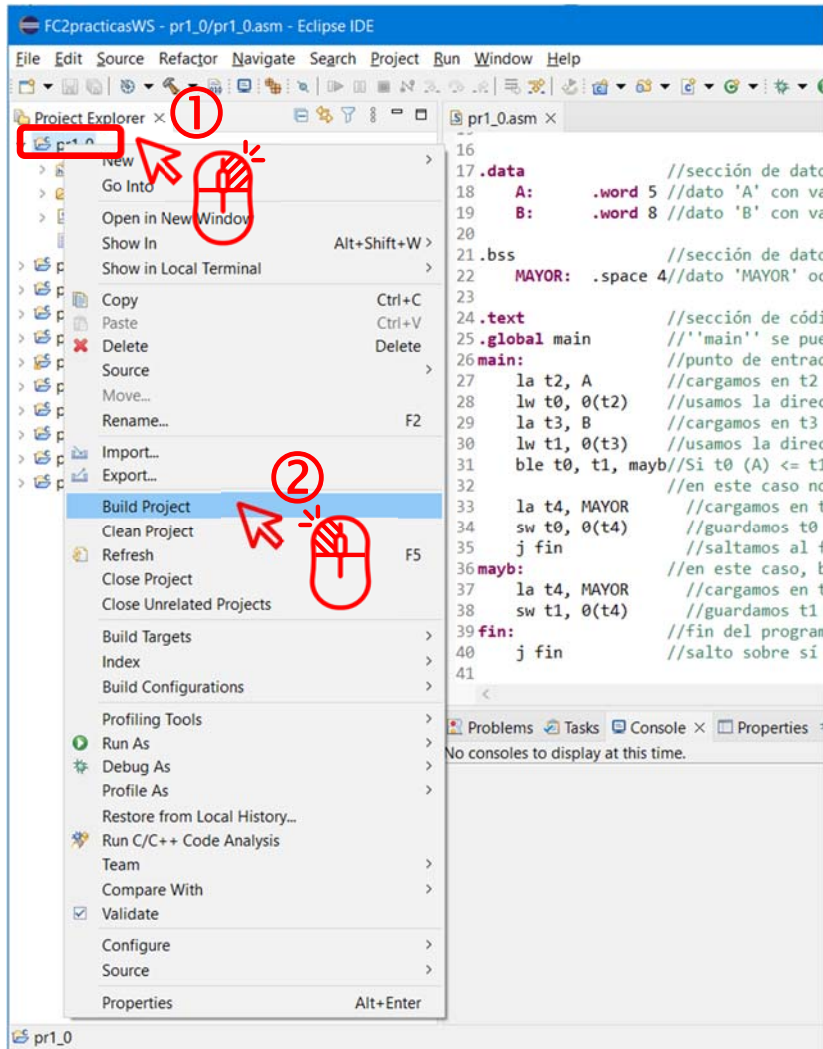
The screenshot shows the Eclipse IDE in the development perspective. The Project Explorer on the left shows a project named 'pr1\_0' with several sub-projects. The file 'pr1\_0.asm' is selected and highlighted with a red box, and a red arrow points to it. The main editor window displays the assembly code for 'pr1\_0.asm', which is also enclosed in a red box. The code includes sections for data, bss, and text, with various instructions and comments. The Outline view on the right shows the structure of the code, including labels 'A', 'B', 'MAYOR', and 'main'. The bottom status bar shows 'Writable', 'Smart Insert', and '1:1:0'.

```
16
17 .data           //sección de datos con valor inicial
18  A:             .word 5 //dato 'A' con valor 5
19  B:             .word 8 //dato 'B' con valor 8
20
21 .bss           //sección de datos sin valor inicial
22  MAYOR:        .space 4//dato 'MAYOR' ocupa 4 bytes
23
24 .text          //sección de código
25 .global main   //'main' se puede llamar desde fuera
26 main:         //punto de entrada del código
27  la t2, A      //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2)  //usamos la dirección para cargar el VALOR de A
29  la t3, B      //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3)  //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb //Si t0 (A) <= t1 (B), salto (condición inversa)
32              //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR  //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4)  //guardamos t0 (A) en MAYOR
35  j fin         //saltamos al final del programa
36 mayb:         //en este caso, b es mayor (o igual)
37  la t4, MAYOR  //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4)  //guardamos t1 (B) en MAYOR
39 fin:          //fin del programa
40  j fin         //salto sobre sí mismo, parando el programa
41
```

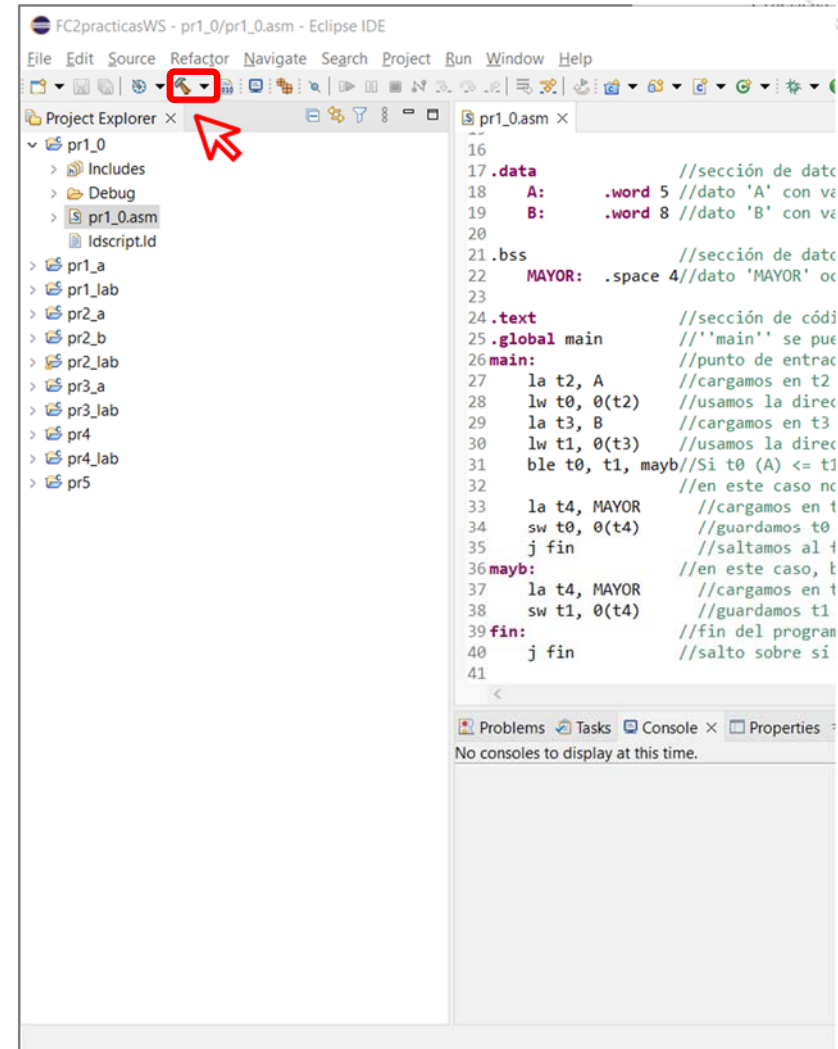


# Development perspective

## Compilation (i)



Option 1



Option 2

# Development perspective

## Compilation (ii)



The screenshot shows the Eclipse IDE interface. The main editor displays assembly code for a program named 'pr1\_0.asm'. The code includes sections for data (.data), uninitialized data (.bss), and text (.text). The main function starts with a 'main' label and performs several operations: loading values from memory, comparing them, and jumping based on the result. The build console at the bottom shows the output of the compilation process, including the invocation of 'riscv64-unknown-elf-size' and the final message: 'Build Finished. 0 errors, 0 warnings: (took 3s.970ms)'. A red box highlights this final message.

```
16
17 .data           //sección de datos con valor inicial
18 A:              .word 5 //dato 'A' con valor 5
19 B:              .word 8 //dato 'B' con valor 8
20
21 .bss            //sección de datos sin valor inicial
22 MAYOR:         .space 4//dato 'MAYOR' ocupa 4 bytes
23
24 .text           //sección de código
25 .global main    //'main' se puede llamar desde fuera
26 main:          //punto de entrada del código
27 la t2, A        //cargamos en t2 la DIRECCIÓN de A
28 lw t0, 0(t2)    //usamos la dirección para cargar el VALOR de A
29 la t3, B        //cargamos en t3 la DIRECCIÓN de B
30 lw t1, 0(t3)    //usamos la dirección para cargar el VALOR de B
31 ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32               //en este caso no hemos saltado (a es mayor)
33 la t4, MAYOR    //cargamos en t4 la DIRECCIÓN de MAYOR
34 sw t0, 0(t4)    //guardamos t0 (A) en MAYOR
35 j fin          //saltamos al final del programa
36 mayb:          //en este caso, b es mayor (o igual)
37 la t4, MAYOR    //cargamos en t4 la DIRECCIÓN de MAYOR
38 sw t1, 0(t4)    //guardamos t1 (B) en MAYOR
39 fin:           //fin del programa
40 j fin          //salto sobre sí mismo, parando el programa
41
```

CDT Build Console [pr1\_0]  
Finished building: pr1\_0.lst'  
..  
'Invoking: GNU RISC-V Cross Print Size'  
riscv64-unknown-elf-size --format=berkeley "pr1\_0.elf"  
text data bss dec hex filename  
60 8 4 72 48 pr1\_0.elf  
'Finished building: pr1\_0.siz'  
..  
16:46:11 Build Finished. 0 errors, 0 warnings: (took 3s.970ms)

Correct compilation



# Development perspective

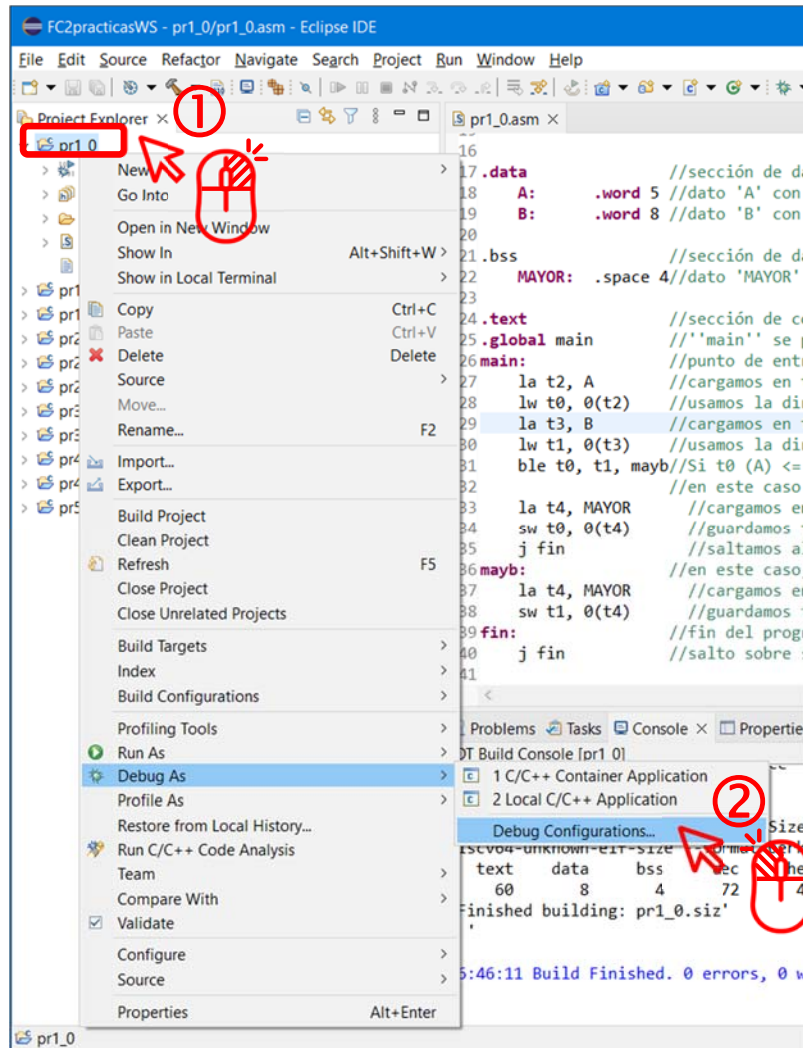
## Debug configurations (i)

15/01/23 version

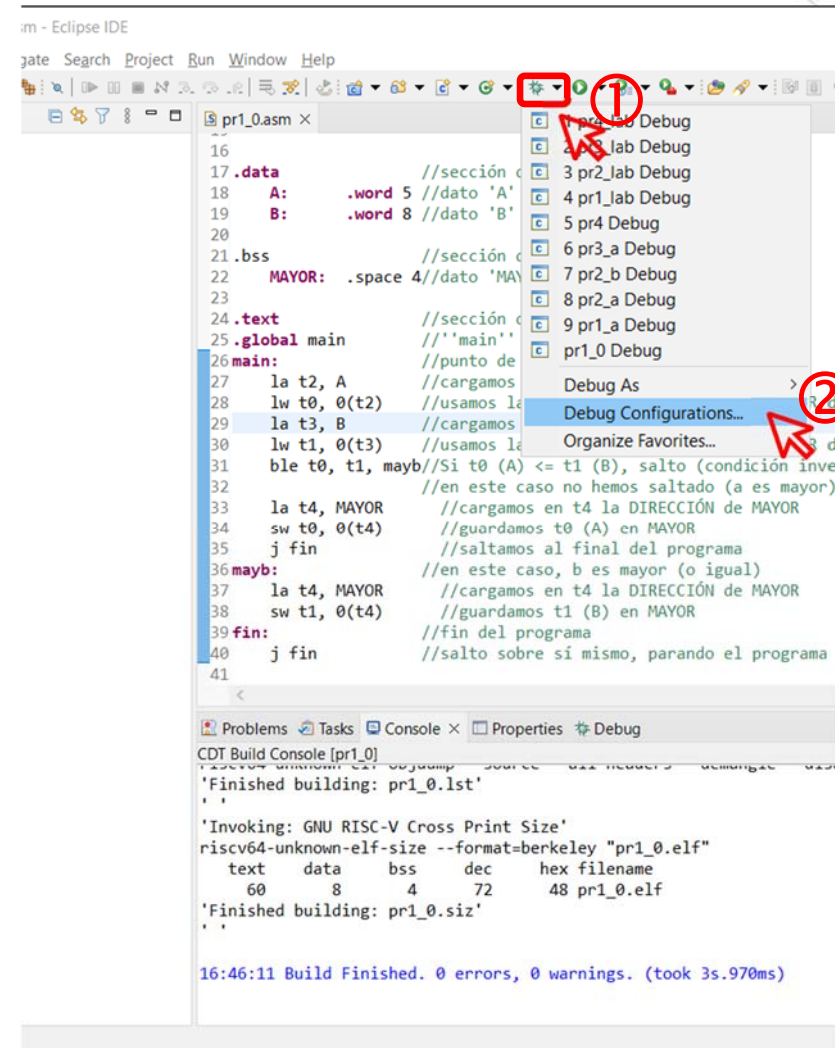
Lab introduction

FC-2

11



Option 1



Option 2

# Development perspective

## Debug configurations (ii)



The screenshot displays the Eclipse IDE interface with the 'Debug Configurations' dialog box open. The dialog is titled 'Create, manage, and run configurations'. On the left, a tree view shows various debug configurations, with 'pr1\_0 Debug' under the 'GDB Hardware Debug' category highlighted by a red box. The main area of the dialog shows the configuration details for 'pr1\_0 Debug'. The 'Name' field is 'pr1\_0 Debug'. The 'Project' is 'pr1\_0'. The 'C/C++ Application' is 'Debug\pr1\_0.elf'. The 'Build (if required) before launching' section is set to 'Select Automatically'. The 'Use workspace settings' radio button is selected. At the bottom right, a red arrow points to the 'Debug' button.





# Development perspective

## Debug configurations (iii)

The screenshot shows the Eclipse IDE interface. The main editor displays the assembly code for `pr1_0.asm`. A dialog box titled "Confirm Perspective Switch" is open, asking the user to switch to the Debug perspective. The dialog includes a question mark icon, a message: "This kind of launch is configured to open the Debug perspective when it suspends. This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints. Switch to this perspective?", a checked checkbox for "Remember my decision", and "Switch" and "No" buttons. A red circle highlights the checkbox, and a red arrow points to the "Switch" button. The background shows the source code and the CDT Build Console output.

```
16
17 .data          //sección de datos con valor inicial
18 A:             .word 5 //dato 'A' con valor 5
19 B:             .word 8 //dato 'B' con valor 8
20
21 .bss           //sección de datos sin valor inicial
22 MAYOR:         .space 4//dato 'MAYOR' ocupa 4 bytes
23
24 .text          //sección de código
25 .globl main
26 main:
27
28
29
30
31
32
33
34
35
36 mayb
37
38
39 fin:           //fin del programa
40 j fin          //salto sobre sí mismo, parando el programa
41
```

CDT Build Console [pr1\_0]  
'Finished building: pr1\_0.lst'  
'Invoking: GNU RISC-V Cross Print Size'  
riscv64-unknown-elf-size --format=berkeley "pr1\_0.elf"  
text data bss dec hex filename  
60 8 4 72 48 pr1\_0.elf  
'Finished building: pr1\_0.siz'  
16:46:11 Build Finished. 0 errors, 0 warnings. (took 3s.970ms)



# Debug perspective

## Main views

The screenshot shows the Eclipse IDE in the Debug perspective. The interface is divided into several panes:

- Project Explorer:** Shows the project structure, including 'pr1\_0.elf' and 'Thread #1 42000 (Suspended: Breakpoint)'.
- Code viewer:** Displays the assembly code for 'pr1\_0.asm'. The current instruction is highlighted: 'la t2, A'. The code includes sections for data, bss, and text.
- Register viewer:** Shows a table with columns 'Name' and 'Value'. A red dashed arrow points to the 'Debug perspective icon' in the top right corner of the IDE.
- Memory viewer:** Shows the output of the 'make all' command in the CDT Build Console, including the command 'riscv64-unknown-elf-size --format=berkeley' and the resulting file sizes for 'pr1\_0.elf'.

Debug perspective icon

Code viewer

Register viewer

Memory viewer



# Debug perspective

## Code view

The screenshot shows the Eclipse IDE interface. The main editor displays assembly code for a program named 'pr1\_0.asm'. The code includes sections for data, bss, and text. The 'main' function is currently selected and highlighted in blue. A red circle with the text 'PC' is positioned over the instruction 'la t2, A' on line 27, indicating the current instruction pointer. The console window at the bottom shows the output of the 'make all' command, including the invocation of 'riscv64-unknown-elf-size' and the resulting file sizes for the 'pr1\_0.elf' file.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:           .word 5 //dato 'A' con valor 5
19  B:           .word 8 //dato 'B' con valor 8
20
21.bss           //sección de datos sin valor inicial
22  MAYOR:      .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text         //sección de código
25.global main  //'main' se puede llamar desde fuera
26main:         //punto de entrada del código
27  la t2, A     //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B     //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32              //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin        //saltamos al final del programa
36mayb:        //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4) //guardamos t1 (B) en MAYOR
39fin:         //fin del programa
40  j fin        //salto sobre sí mismo, parando el programa
41
```

Console output:

```
CDT Build Console [pr1_0]
make all
'Invoking: GNU RISC-V Cross Print Size'
riscv64-unknown-elf-size --format=berkeley "pr1_0.elf"
text  data  bss  dec  hex filename
 60    8    4   72   48 pr1_0.elf
'Finished building: pr1_0.siz'
```





# Debug perspective

## Register view

The screenshot shows the Eclipse IDE in the Debug perspective. The main editor displays assembly code for a program named 'pr1\_0.asm'. The code includes sections for data (.data), bss (.bss), and text (.text). The text section contains the main function, which loads values into registers t2, t3, and t4, and performs a branch based on a comparison between t0 and t1. The Register view on the right shows the General register set, with a red circle highlighting the 'General I' dropdown menu. The register values are as follows:

Name	Value
zero	0
ra	0x0 <main>
sp	0x3ffea80
gp	0x0 <main>
tp	0x0 <main>
t0	0
t1	0
t2	0
fp	0x0 <main>
s1	0
a0	1
a1	0
a2	0
a3	0
a4	0

The Console view at the bottom shows the output of the build process, including the command 'riscv64-unknown-elf-size --format=berkeley "pr1\_0.elf"' and the resulting file sizes for text, data, bss, and dec sections.



# Debug perspective

## Memory view (i)

The screenshot shows the Eclipse IDE interface for debugging assembly code. The main editor displays assembly code for a program named 'pr1\_0.asm'. A 'Monitor Memory' dialog box is open, prompting the user to enter an address or expression to monitor. The address '0x10000' is entered and highlighted with a red box. The dialog box has 'OK' and 'Cancel' buttons, with a red circle and arrow labeled '3' pointing to the 'OK' button. In the bottom right corner, the 'Memory' view is visible, showing a list of memory locations and their values. A red circle and arrow labeled '1' points to the 'Memory' view. In the bottom center, the 'Monitors' toolbar is visible, with a red circle and arrow labeled '2' pointing to it.

```
12**
13**  Notas de diseo:
14**
15** -----*/
16
17 .data          //seccin de datos con valor inicial
18  A:      .word 5 //dato 'A' con valor 5
19  B:      .word 8 //dato 'B' con valor 8
20
21 .bss         //seccin de datos sin valor inicial
22  MAYOR:  .space 4//dato 'MAYOR' ocupa 4 bytes
23
24 .text
25 .global main
26 main:
27  la t2, A
28  lw t0, 0(t2) //carga el valor de A
29  la t3, B
30  lw t1, 0(t3) //carga el valor de B
31  ble t0, t1, mayb //si B es mayor (o igual) que A, salta a mayb
32
33  la t4, MAYOR //carga la direccin de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin //saltamos al final del programa
36 mayb:
37  la t4, MAYOR //cargamos en t4 la DIRECCIN de MAYOR
38  sw t1, 0(t4) //guardamos t1 (B) en MAYOR
39 fin:
40  j fin //fin del programa
41  //salto sobre s mismo, parando el programa
```



# Debug perspective

## Memory view (ii)

The screenshot shows the Eclipse IDE in debug perspective. The main editor displays assembly code for a program. The assembly code includes sections for data, bss, and text. The text section contains the main function, which loads the addresses of variables A and B into registers t2 and t3, compares them, and branches based on the result. The memory view at the bottom shows the address 0x10000, with the value 05000000 highlighted in red. A red dashed arrow points from the assembly code to the memory view.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data //sección de datos con valor inicial
18  A: .word 5 //dato 'A' con valor 5
19  B: .word 8 //dato 'B' con valor 8
20
21.bss //sección de datos sin valor inicial
22  MAYOR: .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text //sección de código
25.global main //''main'' se puede llegar desde fuera
26main: //punto de entrada del código
27  la t2, A //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32 //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin //saltamos al final del programa
36mayb: //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4) //guardamos t1 (B) en MAYOR
39fin: //fin del programa
40  j fin //salto sobre sí mismo, parando el programa
41
```

Address	0 - 3	4 - 7	8 - B	C - F
00010000	05000000	08000000	00000000	00000000
00010010	00000000	00000000	00000000	00000000
00010020	00000000	00000000	00000000	00000000
00010030	00000000	00000000	00000000	00000000
00010040	00000000	00000000	00000000	00000000
00010050	00000000	00000000	00000000	00000000





# Debug perspective

## Step-by-step execution (i)

The screenshot shows the Eclipse IDE in debug perspective for a project named 'pr1\_0'. The main editor displays assembly code for 'pr1\_0.asm'. The code includes data sections for variables A and B, a .bss section for 'MAYOR', and a .text section for the 'main' function. The 'main' function performs several operations: it loads the address of A into register t2, the address of B into register t3, and the address of MAYOR into register t4. It then compares the values in t2 and t3, and branches based on the result. The 'MAYOR' label is defined as a jump to the end of the function.

Annotations in the image include:

- A red circle around the 'Step Into' button in the toolbar.
- A red circle around the 'PC' label in the left margin, with a red arrow pointing to line 29 of the assembly code: `la t3, B`.
- Red boxes around the assembly instructions `A: .word 5` and `la t2, A`.
- A red box around the register 't0' in the 'General' register window, which contains the value 5.
- A red dashed arrow pointing from the 't0' register window to the memory window.

The 'Memory' window at the bottom shows the memory address 0x10000. The value at this address is 05000000, which is highlighted with a red box. This value corresponds to the value of register t0.

Address	0-3	4-7	8-8	C-F
00010000	05000000	00000000	00000000	00000000
00010010	00000000	00000000	00000000	00000000
00010020	00000000	00000000	00000000	00000000
00010030	00000000	00000000	00000000	00000000
00010040	00000000	00000000	00000000	00000000
00010050	00000000	00000000	00000000	00000000



# Debug perspective

## Step-by-step execution (ii)

The screenshot shows the Eclipse IDE in Debug perspective. The main editor displays assembly code for 'pr1\_0.asm'. Key annotations include:

- A red circle around the 'Step' button in the toolbar.
- A red circle around the 'PC' label in the console.
- Red boxes around the code lines: `MAYOR: .space 4`, `la t4, MAYOR`, and `j fin`.
- A red arrow pointing from the `la t4, MAYOR` instruction to the memory view.

The memory view at the bottom shows the address 0x10000. The 8-bit register view shows the value 08000000.

Address	0-3	4-7	8-B	C-F
00010000	05000000	08000000	08000000	00000000
00010010	00000000	00000000	00000000	00000000
00010020	00000000	00000000	00000000	00000000
00010030	00000000	00000000	00000000	00000000
00010040	00000000	00000000	00000000	00000000
00010050	00000000	00000000	00000000	00000000



# Debug perspective

## Format change in the memory view (i)

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure for 'pr1\_0 Debug [GDB Hardware Debugging]'.
- Assembly Editor:** Displays assembly code for 'pr1\_0.asm'. The code includes sections for data (.data), bss (.bss), and text (.text). The text section contains the main function logic.
- Debugger Console:** Shows the execution state, including the current thread and the instruction being executed.
- Memory View:** Displays the memory contents at address 0x10000. The value 05000000 is highlighted in blue, and a red arrow points to it. The memory view is currently set to Hexadecimal format.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:          .word 5 //dato 'A' con valor 5
19  B:          .word 8 //dato 'B' con valor 8
20
21.bss          //sección de datos sin valor inicial
22  MAYOR:     .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text         //sección de código
25.global main  //'main' se puede llamar desde fuera
26.main:       //punto de entrada del código
27  la t2, A    //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B    //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32              //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin       //saltamos al final del programa
36 mayb:       //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4) //guardamos t1 (B) en MAYOR
39 fin:        //fin del programa
40  j fin       //salto sobre si mismo, parando el programa
41
```

Address	0 - 3	4 - 7	8 - B	C - F
00010000	05000000	08000000	08000000	00000000
00010010	00000000	00000000	00000000	00000000
00010020	00000000	00000000	00000000	00000000
00010030	00000000	00000000	00000000	00000000
00010040	00000000	00000000	00000000	00000000
00010050	00000000	00000000	00000000	00000000





# Debug perspective

## Format change in the memory view (ii)

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure for 'pr1\_0 Debug [GDB Hardware Debugging]'.
- Editor:** Displays assembly code for 'pr1\_0.asm'. The code includes sections for data, bss, and text, with instructions like 'la t2, A', 'lw t0, 0(t2)', and 'ble t0, t1, mayb'. Line 40 is highlighted in green.
- Debugger Console:** Shows the 'Memory' view for address 0x10000. A list of renderings is shown, with 'Signed Integer' selected. Red arrows and circles (1 and 2) highlight the selection process.
- Registers:** A table of registers is visible on the right side of the editor.

Name	Value
zero	0
ra	0x0 <main>
sp	0x3ffea80
gp	0x0 <main>
tp	0x0 <main>
t0	5
t1	8
t2	65536
fp	0x0 <main>
s1	0
a0	1
a1	0
a2	0
a3	0
a4	0

Name	Hex	Decimal	Octal	Binary	Default
t2	0x10000	65536	0200000	10000000000000000	65536





# Debug perspective

## Format change in the memory view (iii)

The screenshot shows the Eclipse IDE interface for debugging assembly code. The main editor displays the assembly source code for 'pr1\_0.asm'. The code includes sections for data, bss, and text. The text section contains assembly instructions for loading values into registers, comparing them, and jumping based on the result. The right-hand side shows the 'Registers' window with a list of registers and their values. The 't2' register is highlighted, showing its value as 65536. The bottom window shows the 'Memory' view, which displays a table of memory addresses and their contents. A red box highlights the value '5' in the 0-3 column and '8' in the 8-B column at address 00010000.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:           .word 5 //dato 'A' con valor 5
19  B:           .word 8 //dato 'B' con valor 8
20
21.bss          //sección de datos sin valor inicial
22  MAYOR:      .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text         //sección de código
25.global main  //'main' se puede llamar desde fuera
26main:         //punto de entrada del código
27  la t2, A     //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B     //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32              //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin       //saltamos al final del programa
36mayb:         //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4) //guardamos t1 (B) en MAYOR
39fin:          //fin del programa
40  j fin       //salto sobre si mismo, parando el programa
41
```

Address	0-3	4-7	8-B	C-F
00010000	5	8	A 8	0
00010010	0	0	0	0
00010020	0	0	0	0
00010030	0	0	0	0
00010040	0	0	0	0



# Debug perspective

## Format change in the register view (i)

The screenshot shows the Eclipse IDE in the Debug perspective. The main editor displays assembly code for a program named 'pr1\_0.asm'. The code includes sections for data (.data), bss (.bss), and text (.text). The 'main' function is defined, and the program is currently suspended at line 40, which is a 'j fin' instruction. The Project Explorer on the left shows the project structure. The Register View on the right displays the state of various registers, with 't2' highlighted and containing the value 65536. A context menu is open over the Register View, with the 'Hex' option selected. Red annotations '1' and '2' are present: '1' points to the context menu, and '2' points to the 't2' register entry.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:           .word 5 //dato 'A' con valor 5
19  B:           .word 8 //dato 'B' con valor 8
20
21.bss          //sección de datos sin valor inicial
22  MAYOR:      .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text        //sección de código
25.global main //''main'' se puede llamar desde fuera
26main:        //punto de entrada del código
27  la t2, A    //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B    //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32                //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin       //saltamos al final del programa
36mayb:        //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4) //guardamos t1 (B) en MAYOR
39fin:         //fin del programa
40  j fin       //salto sobre si mismo, parando el programa
41
```

Name	Value
zero	0
ra	0x0 <main>
sp	0x3ffea0
gp	0x0 <main>
tp	0x0 <main>
t0	5
t1	8
t2	65536
fp	0x0 <main>
s1	0

Context Menu Options:

- Select All (Ctrl+A)
- Copy Registers (Ctrl+C)
- Number Format
  - Hex (Selected)
  - Decimal
  - Octal
  - Binary
  - Default
- Find... (Ctrl+F)
- Add Register Group
- Restore Default Register Groups
- Watch
- Restore To Preference

Register View Output:

```
Hex: 00000000
Decimal: 65536
Octal: 0200000
Binary: 10000000000000000
Default: 65536
```



# Debug perspective

## Format change in the register view (ii)

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure for 'pr1\_0 Debug [GDB Hardware Debugging]'.
- Editor:** Displays assembly code for 'pr1\_0.asm'. The code includes sections for data, bss, and text, with instructions like 'la t2, A', 'lw t0, 0(t2)', and 'j fin'.
- Register View:** Located on the right, it shows a list of registers. Register 't2' is highlighted with a red box, and its value is '0x10000 (Hex)'. Below the list, the details for register 't2' are shown: Hex: 0x10000, Decimal: 65536, Octal: 0200000, Binary: 100000000000000000, Default: 65536.
- Debugger Console:** Shows the output of the build process, including 'Finished building: pr1\_0.siz' and '18:38:11 Build Finished. 0 errors, 0 warnings. (took 223ms)'.





# Debug perspective

## Relaunch

The screenshot shows the Eclipse IDE in the Debug perspective. The 'Debug' menu is open, and the 'Terminate and Relaunch' option is highlighted. Red annotations include a circled '1' pointing to the 'Debug' menu, a circled '2' pointing to the 'Terminate and Relaunch' option, and a hand icon pointing to the option. The code editor displays assembly code for 'pr1\_0.asm' with the following content:

```
12**
13** Notas de diseño:
14**
15**-----*/
16
17 .data                //sección de datos con valor inicial
18 A: .word 5           //dato 'A' con valor 5
19 B: .word 8           //dato 'B' con valor 8
20
21 .bss                 //sección de datos sin valor inicial
22 MAYOR: .space 4     //dato 'MAYOR' ocupa 4 bytes
23
24 .text                //sección de código
25 .global main         /*'main' se puede llamar desde fuera
26 main:                //punto de entrada del código
27 la t2, A              //cargamos en t2 la DIRECCIÓN de A
28 lw t0, 0(t2)          //usamos la dirección para cargar el VALOR de A
29 la t3, B              //cargamos en t3 la DIRECCIÓN de B
30 lw t1, 0(t3)          //usamos la dirección para cargar el VALOR de B
31 ble t0, t1, mayb     //Si t0 (A) <= t1 (B), salto (condición inversa)
32                      //en este caso no hemos saltado (a es mayor)
33 la t4, MAYOR          //cargamos en t4 la DIRECCIÓN de MAYOR
34 sw t0, 0(t4)          //guardamos t0 (A) en MAYOR
35 j fin                 //saltamos al final del programa
36 mayb:                //en este caso, b es mayor (o igual)
37 la t4, MAYOR          //cargamos en t4 la DIRECCIÓN de MAYOR
38 sw t1, 0(t4)          //guardamos t1 (B) en MAYOR
39 fin:                 //fin del programa
40 j fin                 //salto sobre si mismo, parando el programa
41
```

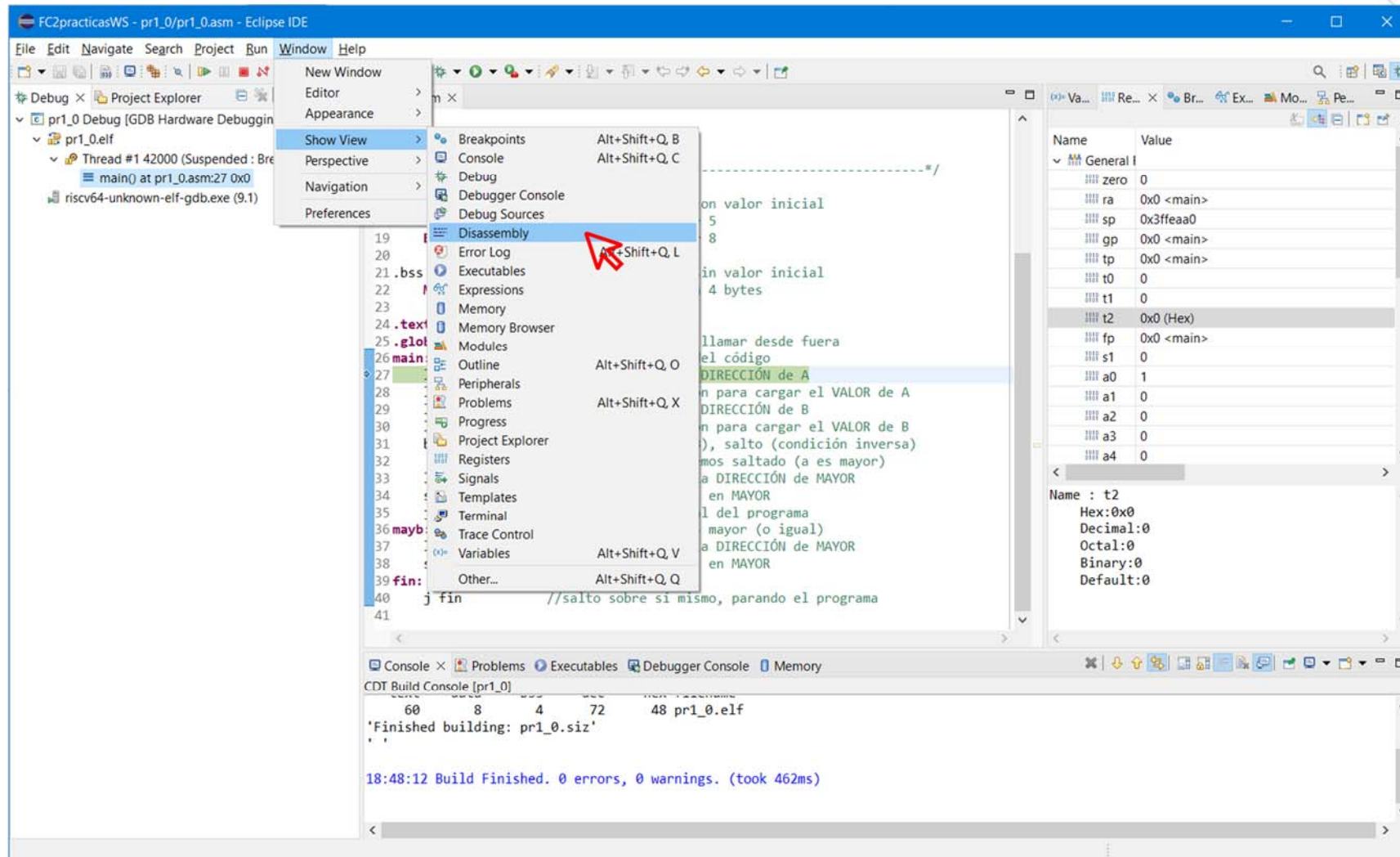
The console shows the following output:

```
CDT Build Console [pr1_0]
60      8      4      72      48 pr1_0.elf
'Finished building: pr1_0.siz'
.
18:38:11 Build Finished. 0 errors, 0 warnings. (took 223ms)
```



# Debug perspective

## Other views: disassembly (i)





# Debug perspective

## Other views: disassembly (ii)

The screenshot displays the Eclipse IDE interface in the Debug perspective. The main editor shows the assembly code for `pr1_0.asm`. The left pane shows the Project Explorer with the current thread and breakpoint information. The right pane shows the disassembly view, highlighting the instruction `auipc t2,0x10` at address `00000000`. The bottom pane shows the build console output, indicating that the build was successful.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17 .data                //sección de datos con valor inicial
18  A:      .word 5 //dato 'A' con valor 5
19  B:      .word 8 //dato 'B' con valor 8
20
21 .bss                //sección de datos sin valor inicial
22  MAYOR:  .space 4//dato 'MAYOR' ocupa 4 bytes
23
24 .text                //sección de código
25 .global main         //"main" se puede llamar desde fuera
26 main:                //punto de entrada del código
27  la t2, A             //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2)        //usamos la dirección para cargar el VALOR de A
29  la t3, B             //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3)        //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb    //Si t0 (A) <= t1 (B), salto (condición inversa)
32                    //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR        //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4)        //guardamos t0 (A) en MAYOR
35  j fin               //saltamos al final del programa
36 mayb:               //en este caso, b es mayor (o igual)
37  la t4, MAYOR        //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4)        //guardamos t1 (B) en MAYOR
39 fin:                //fin del programa
40  j fin               //salto sobre sí mismo, parando el programa
41
```

```
main:
00000000: auipc t2,0x10
00000004: mv t2,t2
28      lw t0, 0(t2) //usamos la
00000008: lw t0,0(t2)
29      la t3, B //cargamos e
0000000c: auipc t3,0x10
00000010: addi t3,t3,-8 # 0x10004
30      lw t1, 0(t3) //usamos la
00000014: lw t1,0(t3)
31      ble t0, t1, mayb//Si t0 (A)
00000018: bge t1,t0,0x2c <mayb>
33      la t4, MAYOR //cargamos
0000001c: auipc t4,0x10
00000020: addi t4,t4,-20 # 0x10008
34      sw t0, 0(t4) //guardamo
00000024: sw t0,0(t4)
35      j fin //saltamos
00000028: j 0x38 <fin>
37      la t4, MAYOR //cargamos
mayb:
0000002c: auipc t4,0x10
00000030: addi t4,t4,-36 # 0x10008
38      sw t1, 0(t4) //guardamo
00000034: sw t1,0(t4)
40      j fin //salto sobr
fin:
00000038: j 0x38 <fin>
0000003c: unimp
```

```
CDT Build Console [pr1_0]
60      8      4      72      48 pr1_0.elf
'Finished building: pr1_0.siz'
.
18:48:12 Build Finished. 0 errors, 0 warnings. (took 462ms)
```





# Debug perspective

## Breakpoints

The screenshot shows the Eclipse IDE interface for debugging assembly code. The main editor displays assembly code for a program named 'pr1\_0.asm'. The code includes sections for data (.data), bss (.bss), and text (.text). A breakpoint is set at line 27, which is highlighted in green. The debugger console at the bottom shows the build process, indicating that the program was built successfully without errors or warnings.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:          .word 5 //dato 'A' con valor 5
19  B:          .word 8 //dato 'B' con valor 8
20
21.bss          //sección de datos sin valor inicial
22  MAYOR:     .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text         //sección de código
25.global main  //'main' se puede llamar desde fuera
26main:         //punto de entrada del código
27  la t2, A    //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B    //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb //Si t0 (A) <= t1 (B), salto (condición inversa)
32  //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin      //saltamos al final del programa
36mayb:        //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4) //guardamos t1 (B) en MAYOR
39fin:         //fin del programa
40  j fin      //salto sobre sí mismo, parando el programa
41
```

Debugger Console [pr1\_0]

```
60      8      4      72      48 pr1_0.elf
'Finished building: pr1_0.siz'
.
18:48:12 Build Finished. 0 errors, 0 warnings. (took 462ms)
```





# Debug perspective

## Execution until a breakpoint

The screenshot shows the Eclipse IDE in the debug perspective. The main editor displays assembly code for a program named 'pr1\_0.asm'. A red circle highlights the play button in the toolbar, and another red circle highlights the 'PC' label next to line 31 in the code editor. The right-hand editor shows the disassembled code with the current PC value highlighted. The bottom console shows the build output.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:          .word 5 //dato 'A' con valor 5
19  B:          .word 8 //dato 'B' con valor 8
20
21.bss          //sección de datos sin valor inicial
22  MAYOR:     .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text         //sección de código
25.global main  //"main" se puede llamar desde fuera
26main:         //punto de entrada del código
27  la t2, A    //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B    //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32              //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin       //saltamos al final del programa
36mayb:        //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4) //guardamos t1 (B) en MAYOR
39fin:         //fin del programa
40  j fin       //salto sobre sí mismo, parando el programa
41
```

Debugger Console:

```
CDT Build Console [pr1_0]
60      8      4      72      48 pr1_0.elf
'Finished building: pr1_0.siz'
.
18:48:12 Build Finished. 0 errors, 0 warnings. (took 462ms)
```



# Debug perspective

## Terminate

The screenshot shows the Eclipse IDE interface in the Debug perspective. The 'Debug' menu is open, and the 'Terminate and Remove' option is highlighted with a red circle and a red arrow labeled '2'. Another red circle and arrow labeled '1' points to the 'pr1\_0 Debug' entry in the Project Explorer. The main editor displays assembly code for a program named 'pr1\_0.asm'. The code includes sections for data, bss, and text, with comments in Spanish. The console at the bottom shows the build output: 'Finished building: pr1\_0.siz' and '19:10:06 Build Finished. 0 errors, 0 warnings. (took 467ms)'. The debugger console is empty, showing 'No debug context'.

# Change of perspective



The screenshot shows the Eclipse IDE interface. The main editor displays assembly code for a program named 'pr1\_0.asm'. The code includes sections for data, bss, and text, with instructions for loading and comparing values. The console at the bottom shows the build process completed successfully. A red circle highlights the 'Run' button in the top toolbar, and a red arrow points to it.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:           .word 5 //dato 'A' con valor 5
19  B:           .word 8 //dato 'B' con valor 8
20
21.bss           //sección de datos sin valor inicial
22  MAYOR:       .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text          //sección de código
25.global main   //'main' se puede llamar desde fuera
26main:          //punto de entrada del código
27  la t2, A      //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2)  //usamos la dirección para cargar el VALOR de A
29  la t3, B      //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3)  //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32              //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR  //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4)  //guardamos t0 (A) en MAYOR
35  j fin        //saltamos al final del programa
36mayb:          //en este caso, b es mayor (o igual)
37  la t4, MAYOR  //cargamos en t4 la DIRECCIÓN de MAYOR
38  sw t1, 0(t4)  //guardamos t1 (B) en MAYOR
39fin:           //fin del programa
40  j fin        //salto sobre sí mismo, parando el programa
41
```

CDT Build Console [pr1\_0]

Address	Disassembly	Comment
60	8	4 72
48 pr1_0.elf		

'Finished building: pr1\_0.siz'

19:10:06 Build Finished. 0 errors, 0 warnings. (took 467ms)





# Development perspective

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows a project named 'pr1\_0' with sub-projects 'pr1\_a' through 'pr1\_5'.
- Editor:** Displays assembly code for 'pr1\_0.asm'. The code includes:

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17 .data                //sección de datos con valor inicial
18  A:      .word 5 //dato 'A' con valor 5
19  B:      .word 8 //dato 'B' con valor 8
20
21 .bss                //sección de datos sin valor inicial
22  MAYOR:  .space 4//dato 'MAYOR' ocupa 4 bytes
23
24 .text                //sección de código
25 .global main        //'main' se puede llamar desde fuera
26 main:               //punto de entrada del código
27  la t2, A            //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2)        //usamos la dirección para cargar el VALOR de A
29  la t3, B            //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3)        //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb    //Si t0 (A) <= t1 (B), salto (condición inversa)
32                    //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR        //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4)        //guardamos t0 (A) en MAYOR
35  j fin              //saltamos al final del programa
36 mayb:               //en este caso, b es mayor (o igual)
37  la t4, MAYOR        //cargamos en t4 la DIRECCIÓN de MAYOR
```
- Outline:** Shows a tree view with nodes for 'A', 'B', 'MAYOR', and 'main'.
- Build Console:** Shows the output of the build process:

```
CDT Build Console [pr1_0]
make all
'Invoking: GNU RISC-V Cross Print Size'
riscv64-unknown-elf-size --format=berkeley "pr1_0.elf"
  text  data  bss  dec  hex filename
   60    8    4   72   48 pr1_0.elf
'Finished building: pr1_0.siz'
.
```



# Workspace change

The screenshot shows the Eclipse IDE interface with the following elements:

- File Menu:** A red circle labeled '1' is around the 'File' menu, with a red arrow pointing to it.
- Switch Workspace:** A red circle labeled '2' is around the 'Switch Workspace' option in the File menu, with a red arrow pointing to it.
- Other...:** A red circle labeled '3' is around the 'Other...' option in the sub-menu, with a red arrow pointing to it.
- Code Editor:** Displays assembly code for 'pr1\_0.asm'. The code includes sections for data, bss, and text, with comments in Spanish. The 'main' function is currently selected.
- Outline:** Shows a project structure with folders for 'A', 'B', 'MAYOR', and 'main'.
- Build Console:** Shows the output of a 'make all' command, including the invocation of 'riscv64-unknown-elf-size' and the final message: '19:10:06 Build Finished. 0 errors, 0 warnings. (took 467ms)'.

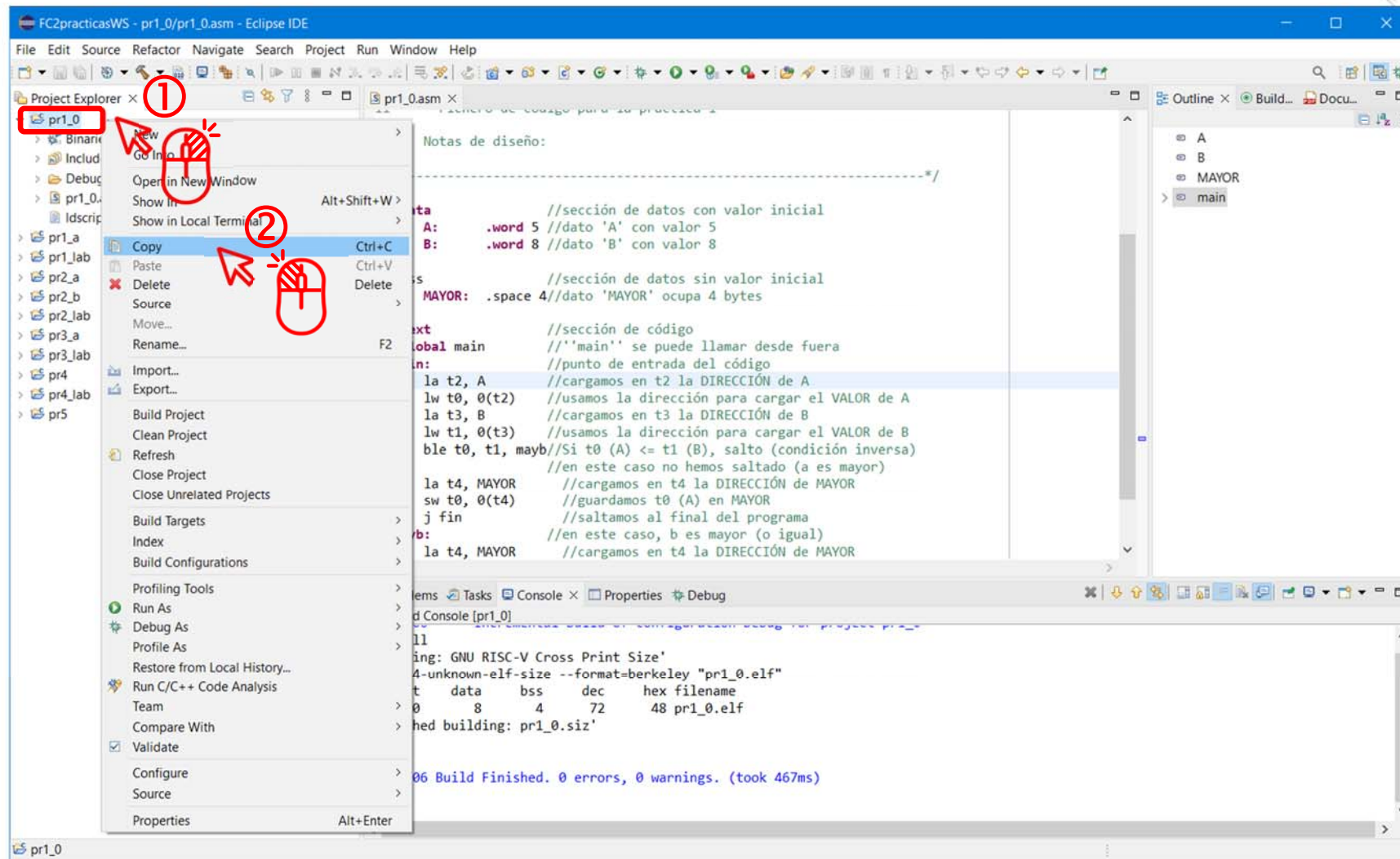


# Creating a new project

## Based on an existing one (i)



15/01/23 version



Copy a project

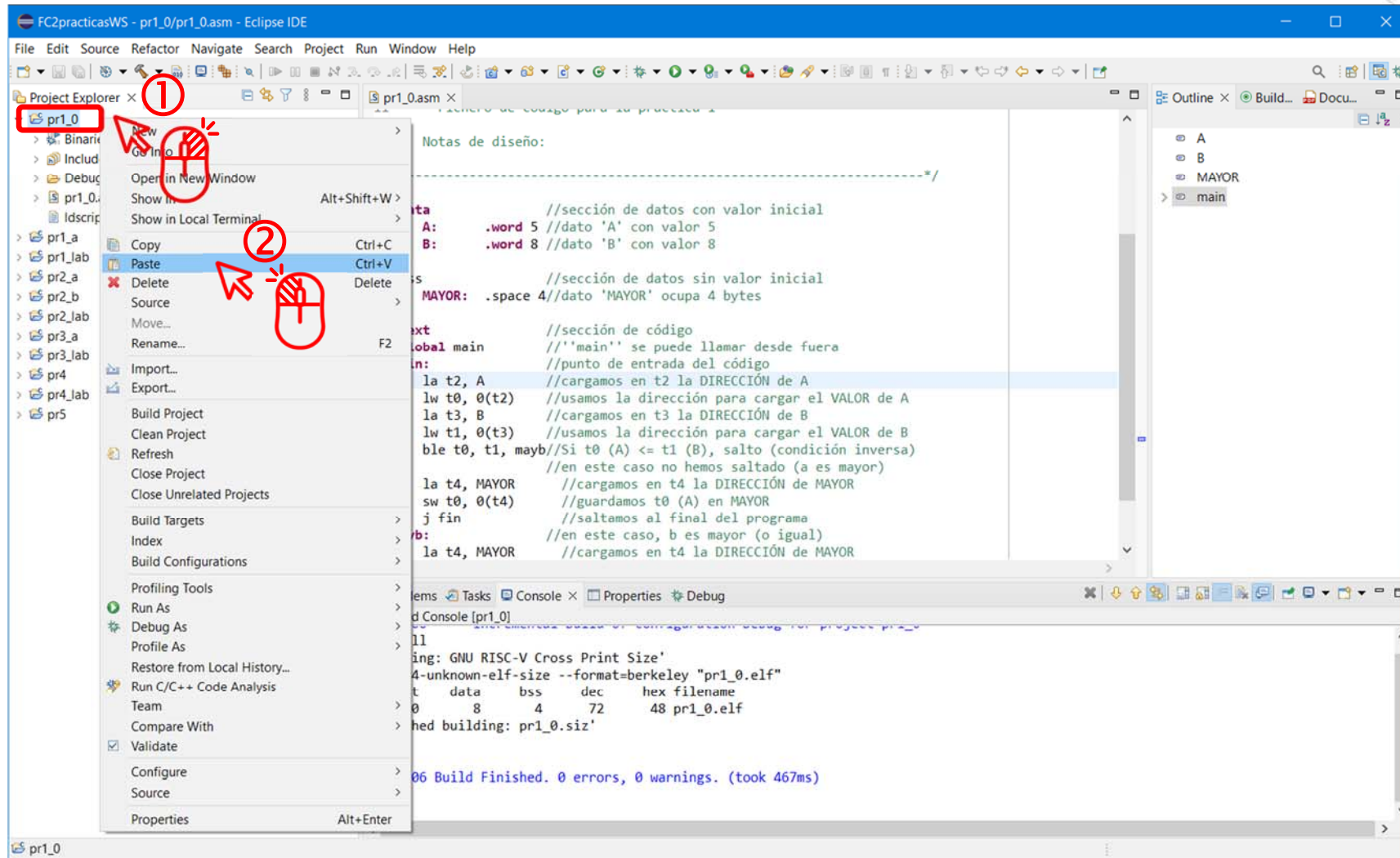




# Creating a new project Based on an existing one (ii)



15/01/23 version



Paste the project





# Creating a new project

## Based on an existing one (iii)



The screenshot shows the Eclipse IDE interface. The main editor displays assembly code for 'pr1\_0.asm'. A 'Copy Project' dialog box is open in the center, with the following fields and options:

- Project name: **nuevoProyecto** (highlighted with a red box)
- Use default location
- Location: D:\FC2practicWS\nuevoProyecto
- Choose file system: default
- Buttons: Copy (highlighted with a red arrow), Cancel

The background assembly code includes:

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:   .word 5 //dato 'A' con valor 5
19  B:   .word 8 //dato 'B' con valor 8
20
21.bss
22  MAYOR: .space
23
24.text
25.global main
26main:
27  la t2, A
28  lw t0, 0(t2)
29  la t3, B
30  lw t1, 0(t3)
31  ble t0, t1, may
32
33  la t4, MAYOR
34  sw t0, 0(t4)
35  j fin
36mayb:
37  la t4, MAYOR //saltamos al final del programa
                //en este caso, b es mayor (o igual)
                //cargamos en t4 la DIRECCIÓN de MAYOR
```

The build console at the bottom shows the following output:

```
CDT Build Console [pr1_0]
make all
'Invoking: GNU RISC-V Cross Print Size'
riscv64-unknown-elf-size --format=berkeley "pr1_0.elf"
text data bss dec hex filename
60 8 4 72 48 pr1_0.elf
'Finished building: pr1_0.siz'
19:10:06 Build Finished. 0 errors, 0 warnings. (took 467ms)
```



# Creating a new project Based on an existing one (iv)



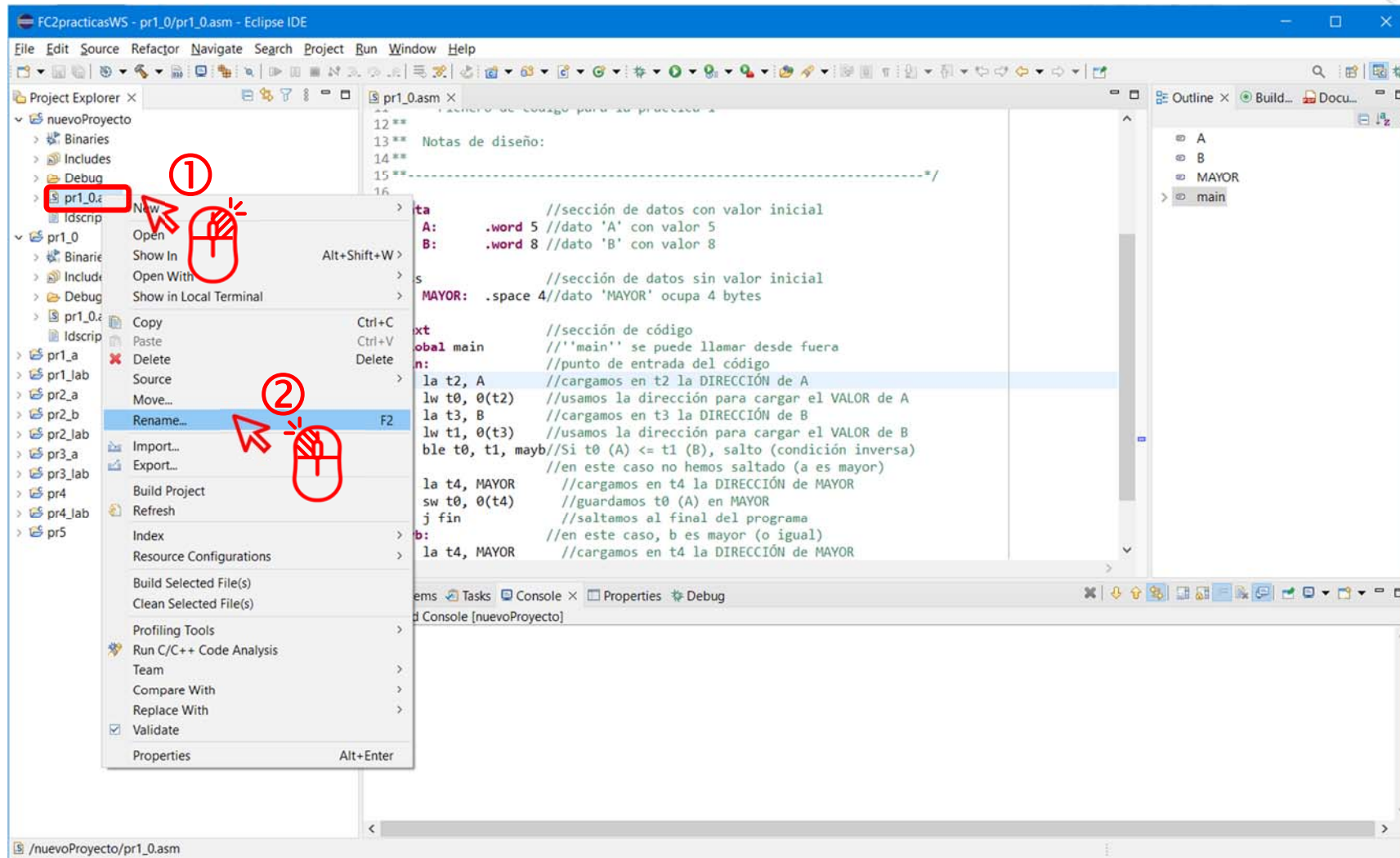
The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer shows a tree view with a new project named 'nuevoProyecto' highlighted by a red circle. Below it, several sub-projects are listed, including 'pr1\_0' and 'pr1\_0.asm'. The main editor window displays the assembly code for 'pr1\_0.asm'. The code includes comments in Spanish and assembly instructions for data, BSS, and text sections. The text section contains a 'main' function that compares two values 'A' and 'B' and prints the larger one. The status bar at the bottom indicates the current file path as '/nuevoProyecto/pr1\_0.asm'.

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:           .word 5 //dato 'A' con valor 5
19  B:           .word 8 //dato 'B' con valor 8
20
21.bss          //sección de datos sin valor inicial
22  MAYOR:      .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text        //sección de código
25.global main //''main'' se puede llamar desde fuera
26main:        //punto de entrada del código
27  la t2, A    //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B    //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb //Si t0 (A) <= t1 (B), salto (condición inversa)
32              //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin       //saltamos al final del programa
36mayb:        //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
```



# Creating a new project

## Based on an existing one (v)



Rename the source code



# Creating a new project

## Based on an existing one (vi)



The screenshot shows the Eclipse IDE interface. The 'Project Explorer' on the left shows a project named 'nuevoProyecto' with subfolders 'Binaries', 'Includes', 'Debug', and 'pr1\_0.asm'. Below it is another project 'pr1\_0' with similar subfolders. The 'Editor' window shows the assembly code for 'pr1\_0.asm'. A 'Rename Resource' dialog box is open, with the 'New name' field containing 'nuevoProyecto.asm'. The 'Update references' checkbox is checked. A red arrow points to the 'OK' button. The 'Outline' view on the right shows the project structure with 'main' selected. The 'Problems' and 'Tasks' views are empty. The 'Console' view at the bottom shows 'CDT Build Console [nuevoProyecto]'.





# Creating a new project Based on an existing one (vii)



15/01/23 version

```
12**
13**  Notas de diseño:
14**
15**-----*/
16
17.data          //sección de datos con valor inicial
18  A:           .word 5 //dato 'A' con valor 5
19  B:           .word 8 //dato 'B' con valor 8
20
21.bss          //sección de datos sin valor inicial
22  MAYOR:      .space 4//dato 'MAYOR' ocupa 4 bytes
23
24.text        //sección de código
25.global main //''main'' se puede llamar desde fuera
26main:        //punto de entrada del código
27  la t2, A    //cargamos en t2 la DIRECCIÓN de A
28  lw t0, 0(t2) //usamos la dirección para cargar el VALOR de A
29  la t3, B    //cargamos en t3 la DIRECCIÓN de B
30  lw t1, 0(t3) //usamos la dirección para cargar el VALOR de B
31  ble t0, t1, mayb//Si t0 (A) <= t1 (B), salto (condición inversa)
32              //en este caso no hemos saltado (a es mayor)
33  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
34  sw t0, 0(t4) //guardamos t0 (A) en MAYOR
35  j fin       //saltamos al final del programa
36mayb:        //en este caso, b es mayor (o igual)
37  la t4, MAYOR //cargamos en t4 la DIRECCIÓN de MAYOR
```

Lab introduction

FC-2

41

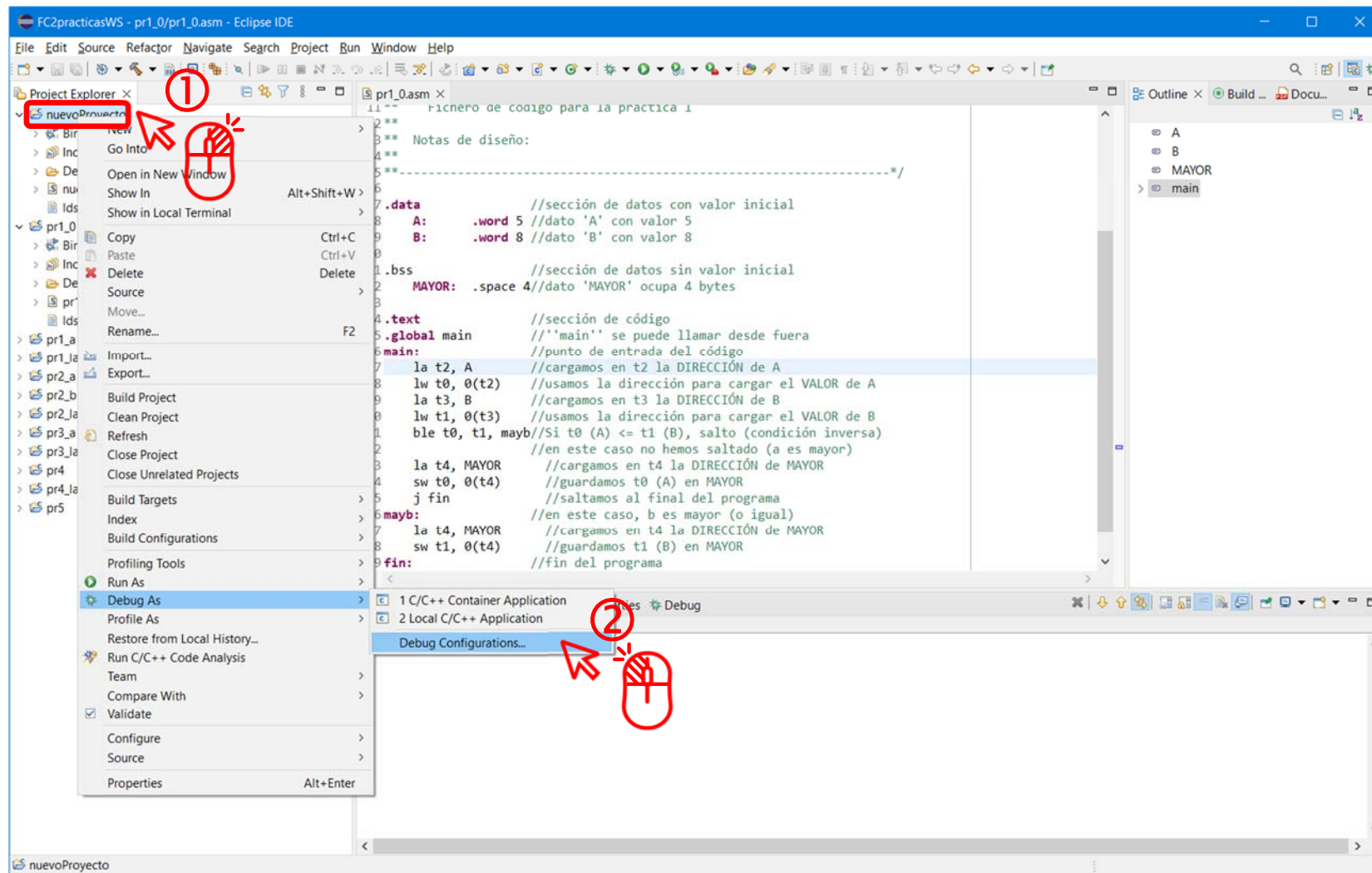


# Creating a new project

## Based on an existing one (vii)



15/01/23 version



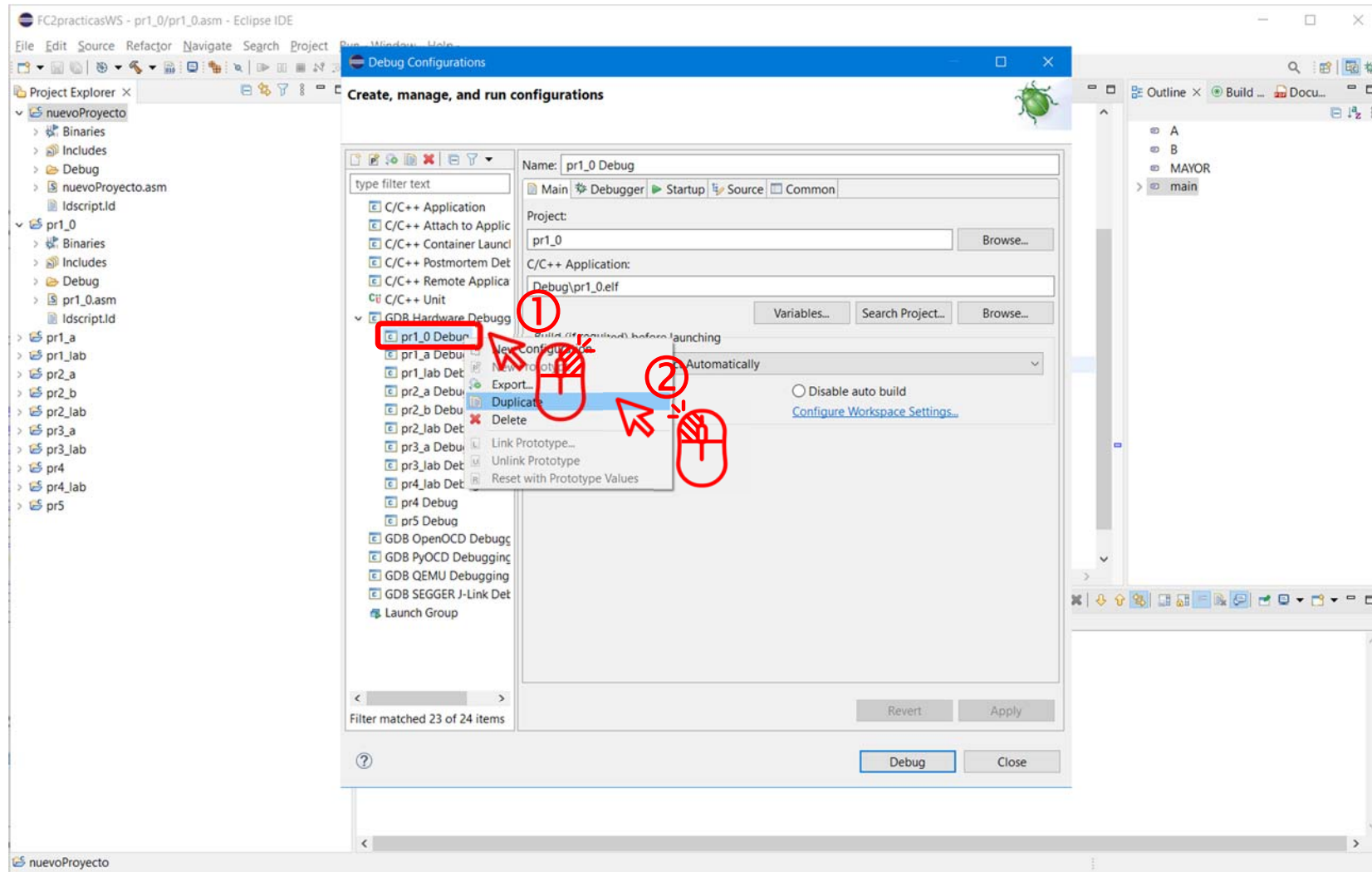
Lab introduction

FC-2

42



# Creating a new project Based on an existing one (viii)

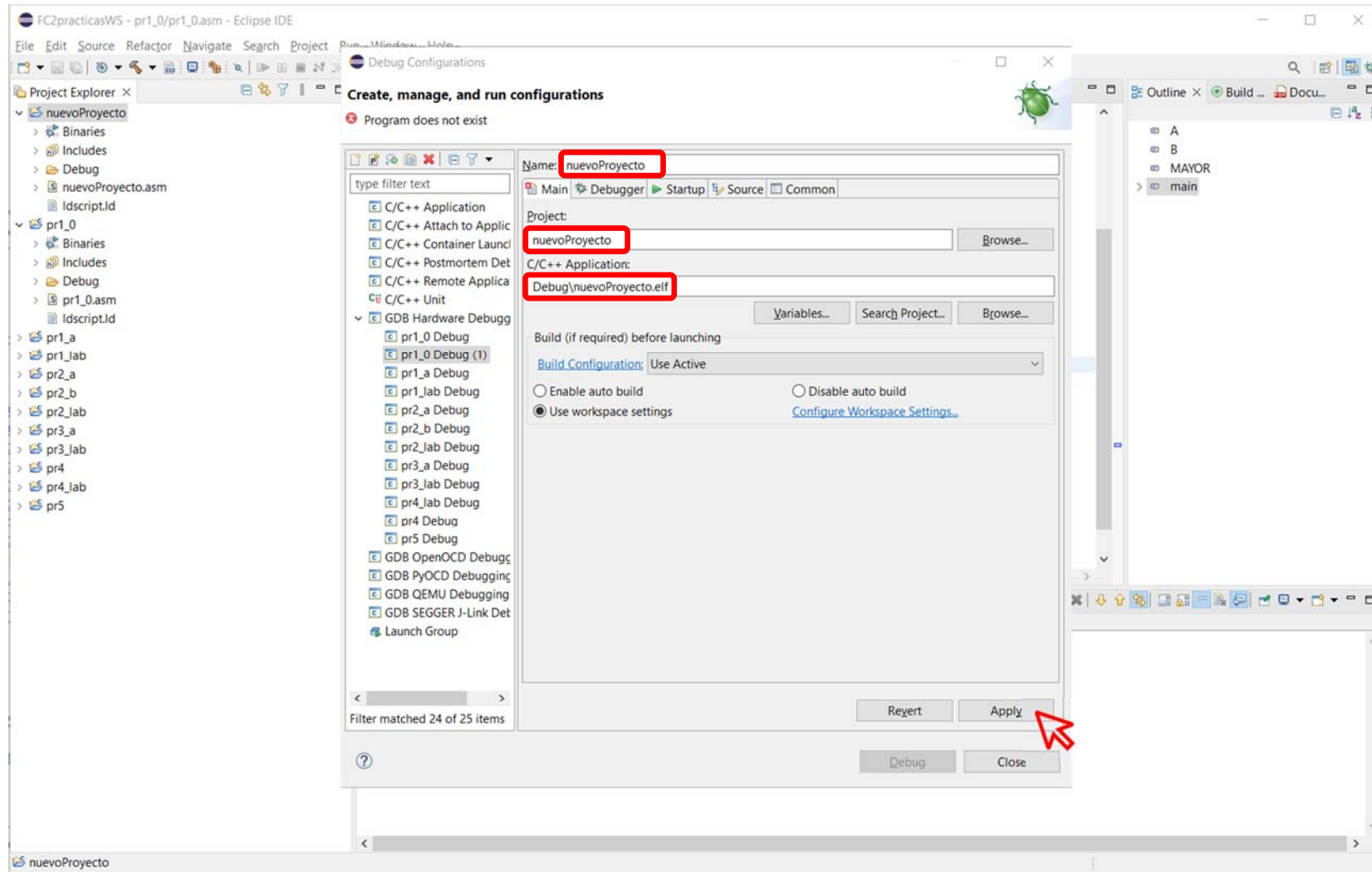


Duplicate the debug configuration



# Creating a new project

## Based on an existing one (ix)



Rename the debug configuration





# Creating a new project Based on an existing one (x)



The screenshot shows the Eclipse IDE interface. The 'Create, manage, and run configurations' dialog is open, displaying a list of configurations on the left. Under the 'GDB Hardware Debug' category, 'nuevoProyecto' is selected and highlighted with a red box. The main area of the dialog shows configuration details for 'nuevoProyecto', including the project name, location, and build settings. The 'Build Configuration' dropdown is set to 'Use Active'. The 'Debug' button is visible at the bottom of the dialog.

# About *Creative Commons*



## ■ CC license (*Creative Commons*)

- This license enables reusers to distribute, remix, adapt, and build upon the material in any medium or format for noncommercial purposes only, and only so long as attribution is given to the creator. If you remix, adapt, or build upon the material, you must license the modified material under identical terms:



### **Attribution:**

Credit must be given to the creator.



### **Non commercial:**

Only noncommercial uses of the work are permitted.



### **Share alike:**

Adaptations must be shared under the same terms.

**More information:** <https://creativecommons.org/licenses/by-nc-sa/4.0/>