

Sistemas Operativos

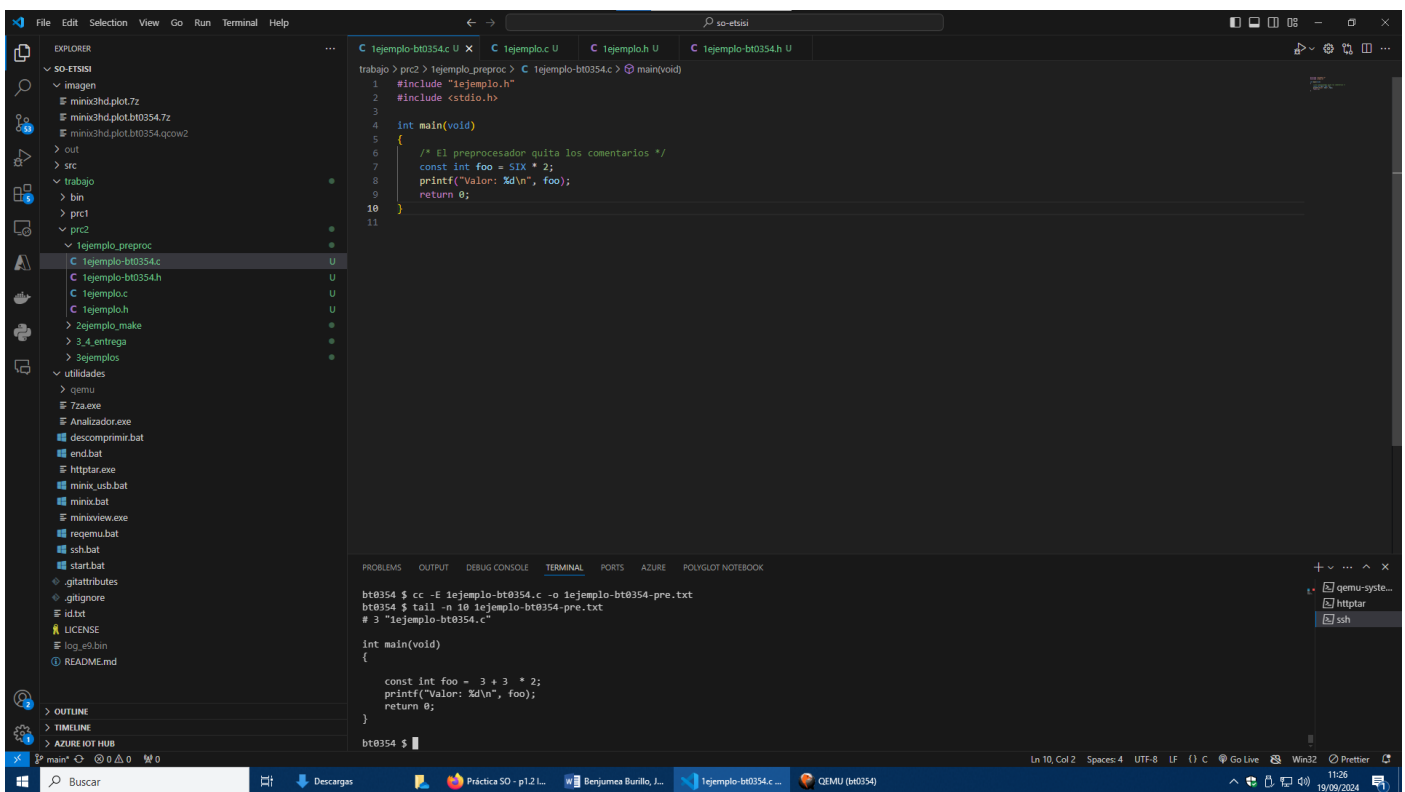
Práctica 1.2 – Introducción a C

Fecha: 19/09/2024

Alumno: Jorge Benjumea Burillo

Correo: jorge.benjumea@alumnos.upm.es

1. Preprocesador



The screenshot shows the Visual Studio Code interface with a C program being edited and compiled. The Explorer panel on the left shows the project structure, including files like `1ejemplo_preproc`, `1ejemplo-bt0354.c`, `1ejemplo-bt0354.h`, `1ejemplo.c`, and `1ejemplo.h`. The main editor displays the following C code:

```
trabajo > prc2 > 1ejemplo_preproc > C 1ejemplo-bt0354.c > main(void)
1 #include "1ejemplo.h"
2 #include <stdio.h>
3
4 int main(void)
5 {
6     /* El preprocesador quita los comentarios */
7     const int foo = SIX * 2;
8     printf("Valor: %d\n", foo);
9     return 0;
10 }
11
```

The TERMINAL panel at the bottom shows the execution of the preprocessor and compiler:

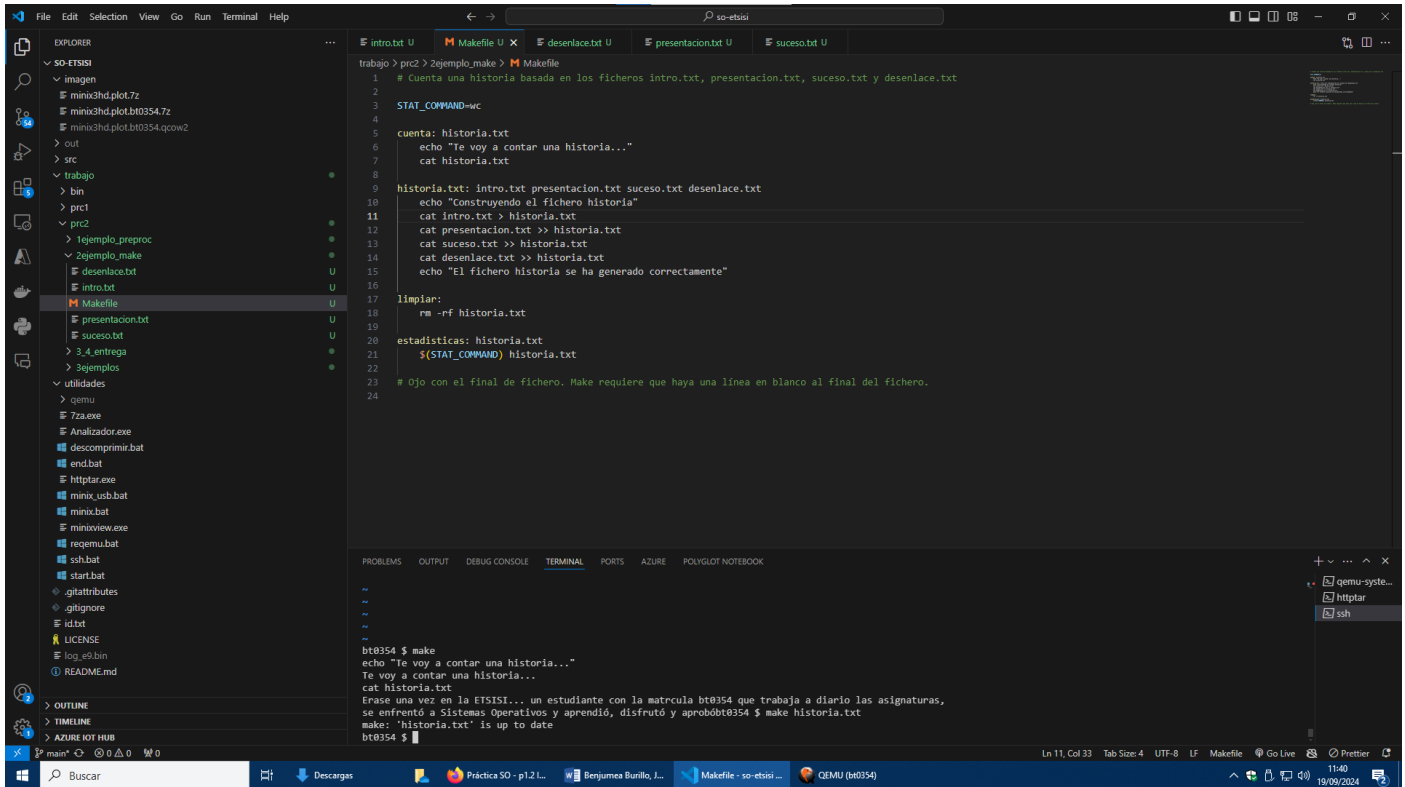
```
bt0354 $ cc -E 1ejemplo-bt0354.c -o 1ejemplo-bt0354-pre.txt
bt0354 $ tail -n 10 1ejemplo-bt0354-pre.txt
# 3 "1ejemplo-bt0354.c"

int main(void)
{
    const int foo = 3 + 3 * 2;
    printf("Valor: %d\n", foo);
    return 0;
}

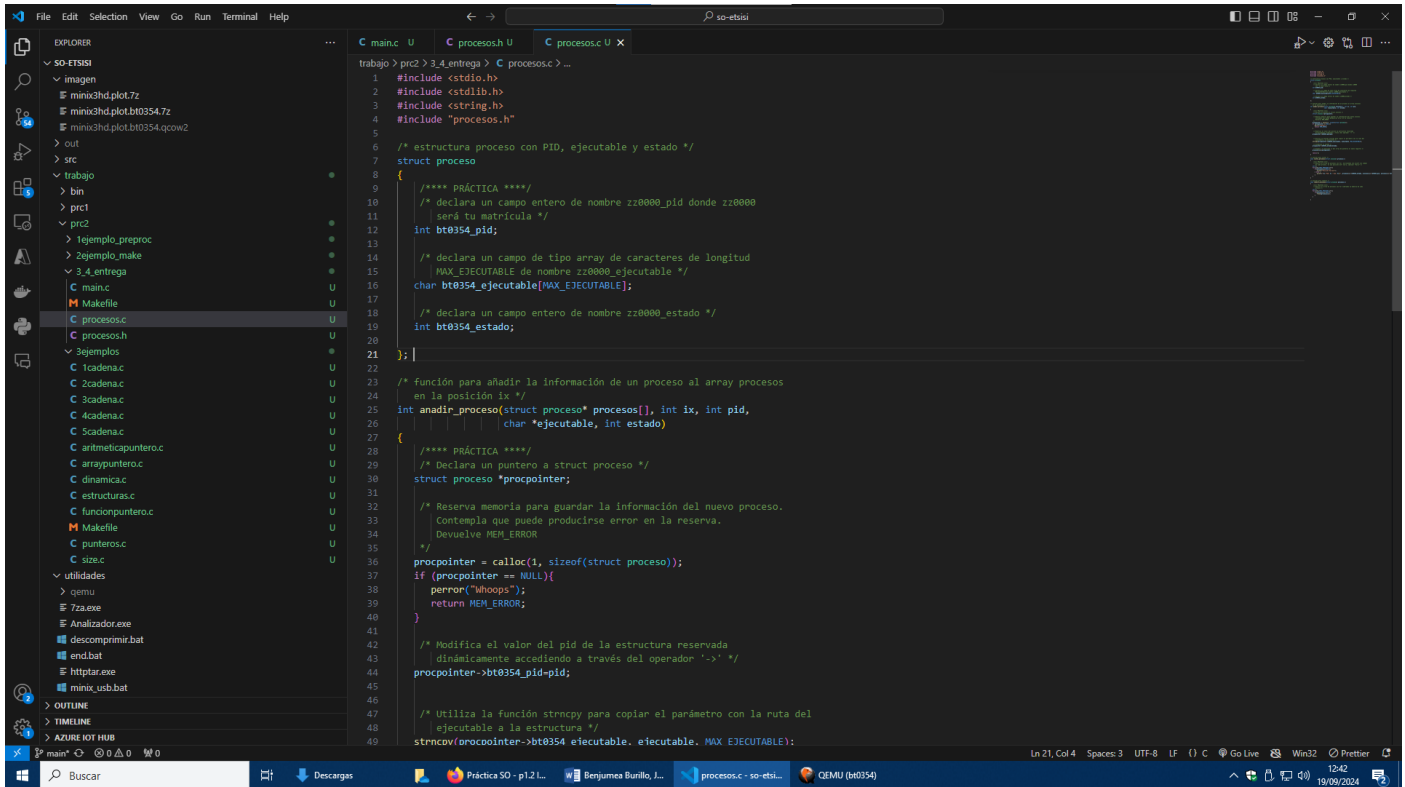
bt0354 $
```

The status bar at the bottom indicates the current file is `1ejemplo-bt0354.c` and the editor is using the C99 standard.

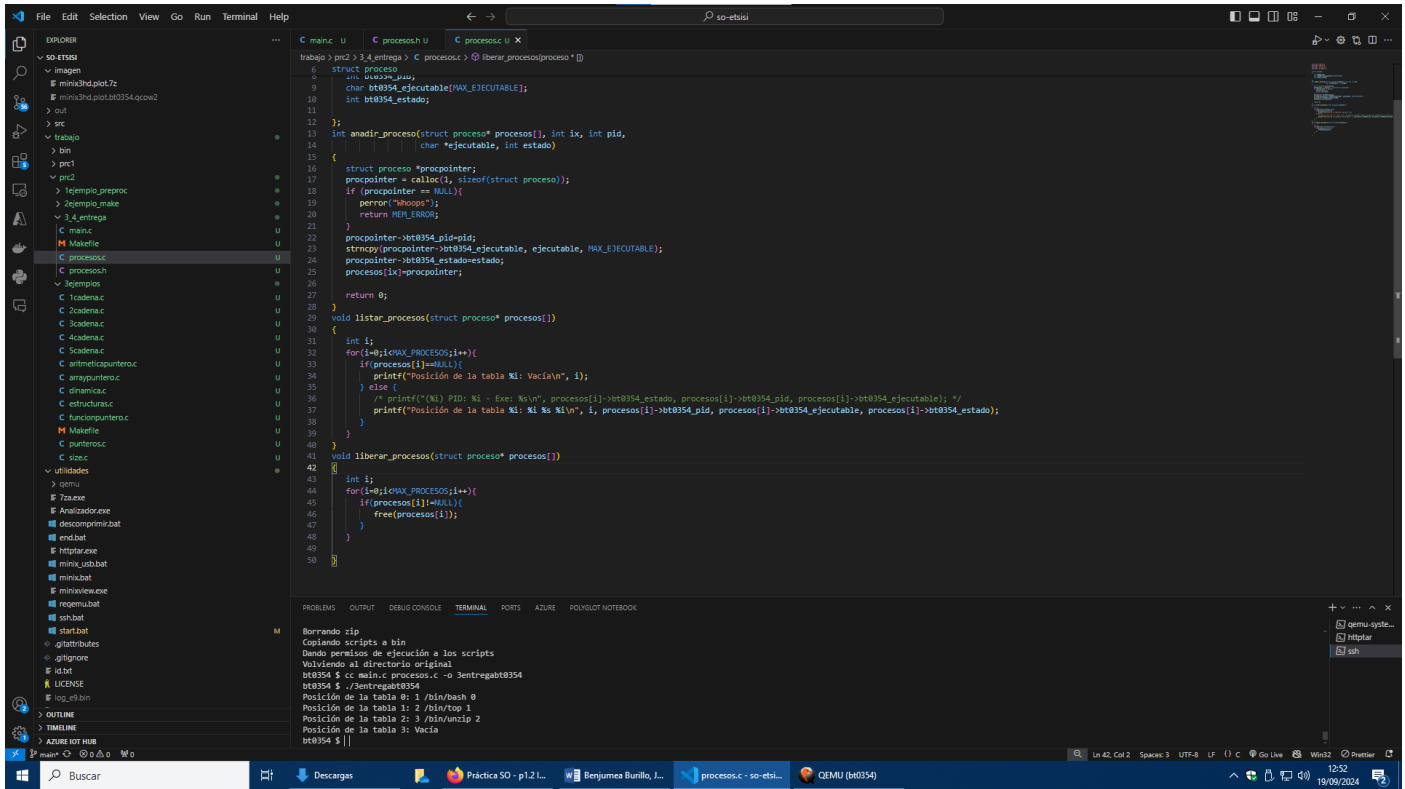
2. Introducción a make



3. Estructura C para gestión proceso



4. Funciones C para gestión de array



```
6 struct proceso
7 {
8     char *ejecutable;
9     char bt0354_ejecutable[MAX_EJECUTABLE];
10    int bt0354_estado;
11 };
12
13 int anadir_proceso(struct proceso* procesos[], int ix, int pid,
14                  char *ejecutable, int estado)
15 {
16     struct proceso *procpunter;
17     procpunter = calloc(1, sizeof(struct proceso));
18     if (procpunter == NULL){
19         perror("malloc");
20         return MEH_ERROUR;
21     }
22     procpunter->bt0354_pid=pid;
23     strncpy(procpunter->bt0354_ejecutable, ejecutable, MAX_EJECUTABLE);
24     procpunter->bt0354_estado=estado;
25     procesos[ix]=procpunter;
26
27     return 0;
28 }
29
30 void listar_procesos(struct proceso* procesos[])
31 {
32     int i;
33     for(i=0; i<MAX_PROCESOS; i++){
34         if(procesos[i]!=NULL){
35             printf("Posición de la tabla %i: Vacía\n", i);
36         } else {
37             printf("(%) PID: %i - Ejec: %s\n", procesos[i]->bt0354_estado, procesos[i]->bt0354_pid, procesos[i]->bt0354_ejecutable);
38             printf("Posición de la tabla %i: %i %s %i\n", i, procesos[i]->bt0354_pid, procesos[i]->bt0354_ejecutable, procesos[i]->bt0354_estado);
39         }
40     }
41 }
42
43 void liberar_procesos(struct proceso* procesos[])
44 {
45     int i;
46     for(i=0; i<MAX_PROCESOS; i++){
47         if(procesos[i]!=NULL){
48             free(procesos[i]);
49         }
50     }
51 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE POLYGLOT NOTEBOOK

Borrando zip
Copiando scripts a bin
Dando permisos de ejecución a los scripts
Volviendo al directorio original
bt0354 \$ cc main.c procesos.c -o 3entregabt0354
bt0354 \$./3entregabt0354
Posición de la tabla 0: 1 /bin/bash 0
Posición de la tabla 1: 2 /bin/top 1
Posición de la tabla 2: 3 /bin/unzip 2
Posición de la tabla 3: Vacía
bt0354 \$ ||

Ln 42, Col 2 Spaces: 3 UTF-8 LF C Go Live Wini32 Previews

12:32 18/09/2024

5. Makefile

