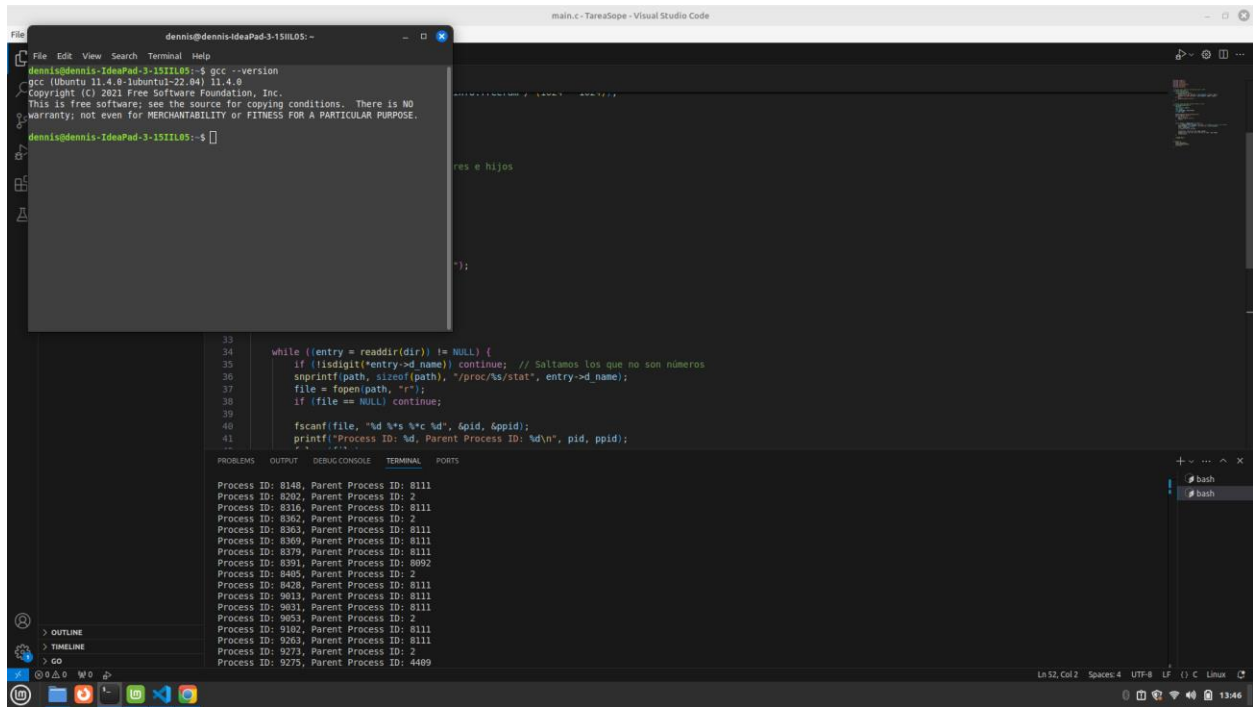


TAREA 1

La instalación del modulo



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the command `gcc --version` and its output, which includes the version number 11.4.0 and copyright information. The code editor in the background shows a C program with a `while` loop that reads directory entries and prints process information.

```
dennis@dennis-IdeaPad-3-1511L05:~$ gcc --version
gcc (Ubuntu 11.4.0-1ubuntu1-22.04) 11.4.0
Copyright (c) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

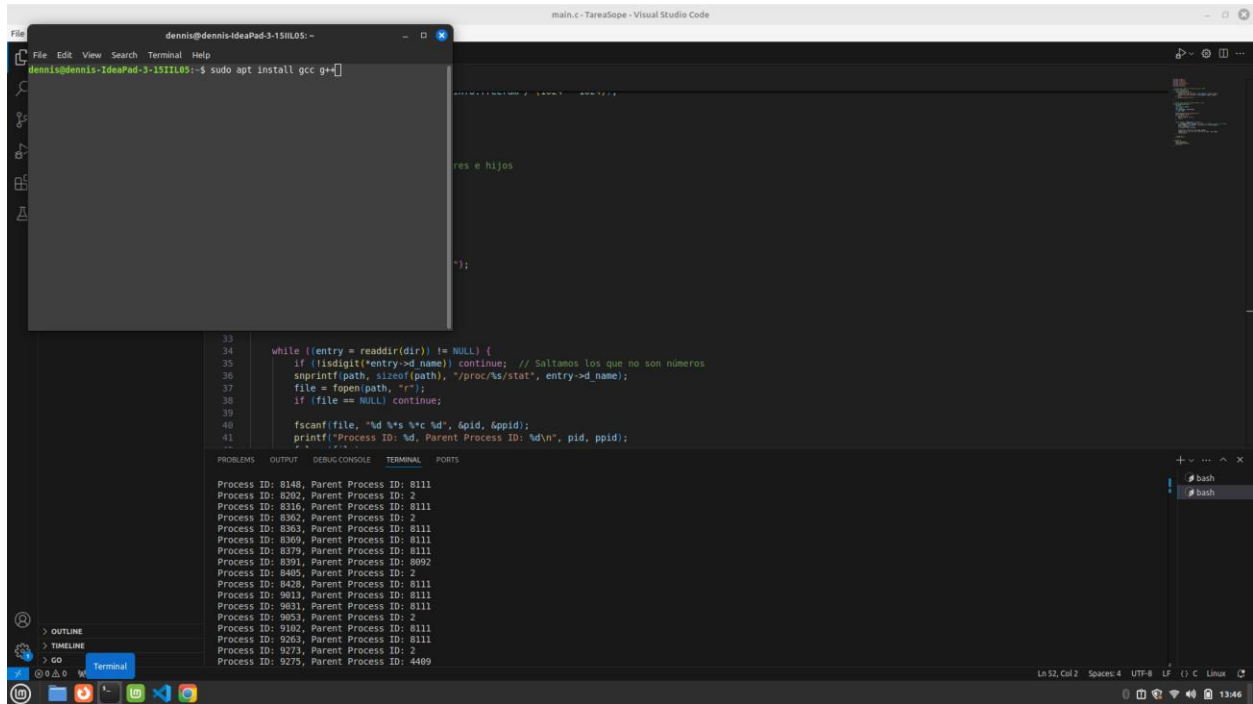
dennis@dennis-IdeaPad-3-1511L05:~$
```

```
33
34 while ((entry = readdir(dir)) != NULL) {
35     if (!isdigit(entry->d_name)) continue; // Saltamos los que no son números
36     snprintf(path, sizeof(path), "/proc/%s/stat", entry->d_name);
37     file = fopen(path, "r");
38     if (file == NULL) continue;
39
40     fscanf(file, "%d %s %c %d", &pid, &ppid);
41     printf("Process ID: %d, Parent Process ID: %d\n", pid, ppid);
42 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Process ID: 8148, Parent Process ID: 8111
Process ID: 8202, Parent Process ID: 2
Process ID: 8316, Parent Process ID: 8111
Process ID: 8362, Parent Process ID: 2
Process ID: 8363, Parent Process ID: 8111
Process ID: 8369, Parent Process ID: 8111
Process ID: 8379, Parent Process ID: 8111
Process ID: 8391, Parent Process ID: 8092
Process ID: 8405, Parent Process ID: 2
Process ID: 8428, Parent Process ID: 8111
Process ID: 9013, Parent Process ID: 8111
Process ID: 9031, Parent Process ID: 8111
Process ID: 9053, Parent Process ID: 2
Process ID: 9102, Parent Process ID: 8111
Process ID: 9263, Parent Process ID: 8111
Process ID: 9273, Parent Process ID: 2
Process ID: 9275, Parent Process ID: 4409
```

Ln 52, Col 2 Spaces: 4 UTF-8 LF () C Linux 13:46



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the command `sudo apt install gcc g++` and its output, which includes the version number 11.4.0 and copyright information. The code editor in the background shows a C program with a `while` loop that reads directory entries and prints process information.

```
dennis@dennis-IdeaPad-3-1511L05:~$ sudo apt install gcc g++
gcc (Ubuntu 11.4.0-1ubuntu1-22.04) 11.4.0
Copyright (c) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

dennis@dennis-IdeaPad-3-1511L05:~$
```

```
33
34 while ((entry = readdir(dir)) != NULL) {
35     if (!isdigit(entry->d_name)) continue; // Saltamos los que no son números
36     snprintf(path, sizeof(path), "/proc/%s/stat", entry->d_name);
37     file = fopen(path, "r");
38     if (file == NULL) continue;
39
40     fscanf(file, "%d %s %c %d", &pid, &ppid);
41     printf("Process ID: %d, Parent Process ID: %d\n", pid, ppid);
42 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Process ID: 8148, Parent Process ID: 8111
Process ID: 8202, Parent Process ID: 2
Process ID: 8316, Parent Process ID: 8111
Process ID: 8362, Parent Process ID: 2
Process ID: 8363, Parent Process ID: 8111
Process ID: 8369, Parent Process ID: 8111
Process ID: 8379, Parent Process ID: 8111
Process ID: 8391, Parent Process ID: 8092
Process ID: 8405, Parent Process ID: 2
Process ID: 8428, Parent Process ID: 8111
Process ID: 9013, Parent Process ID: 8111
Process ID: 9031, Parent Process ID: 8111
Process ID: 9053, Parent Process ID: 2
Process ID: 9102, Parent Process ID: 8111
Process ID: 9263, Parent Process ID: 8111
Process ID: 9273, Parent Process ID: 2
Process ID: 9275, Parent Process ID: 4409
```

Ln 52, Col 2 Spaces: 4 UTF-8 LF () C Linux 13:46

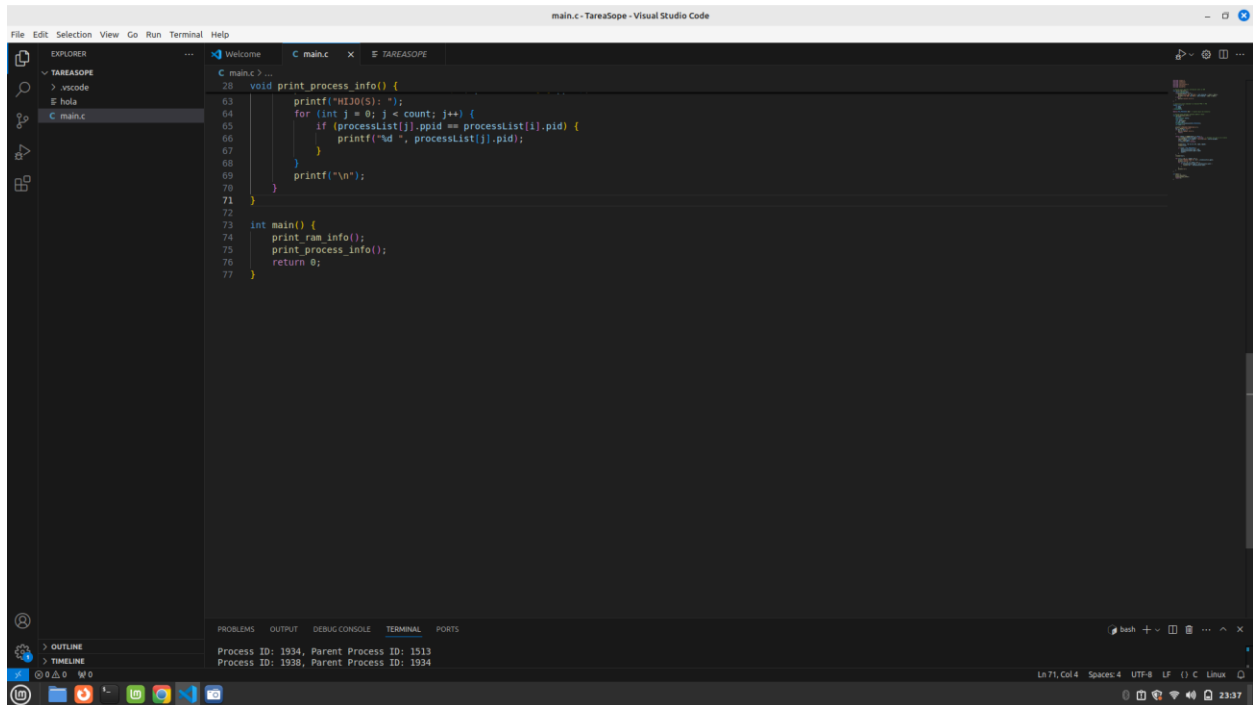
TAREA 1

Salida

[illegible][illegible]

TAREA 1

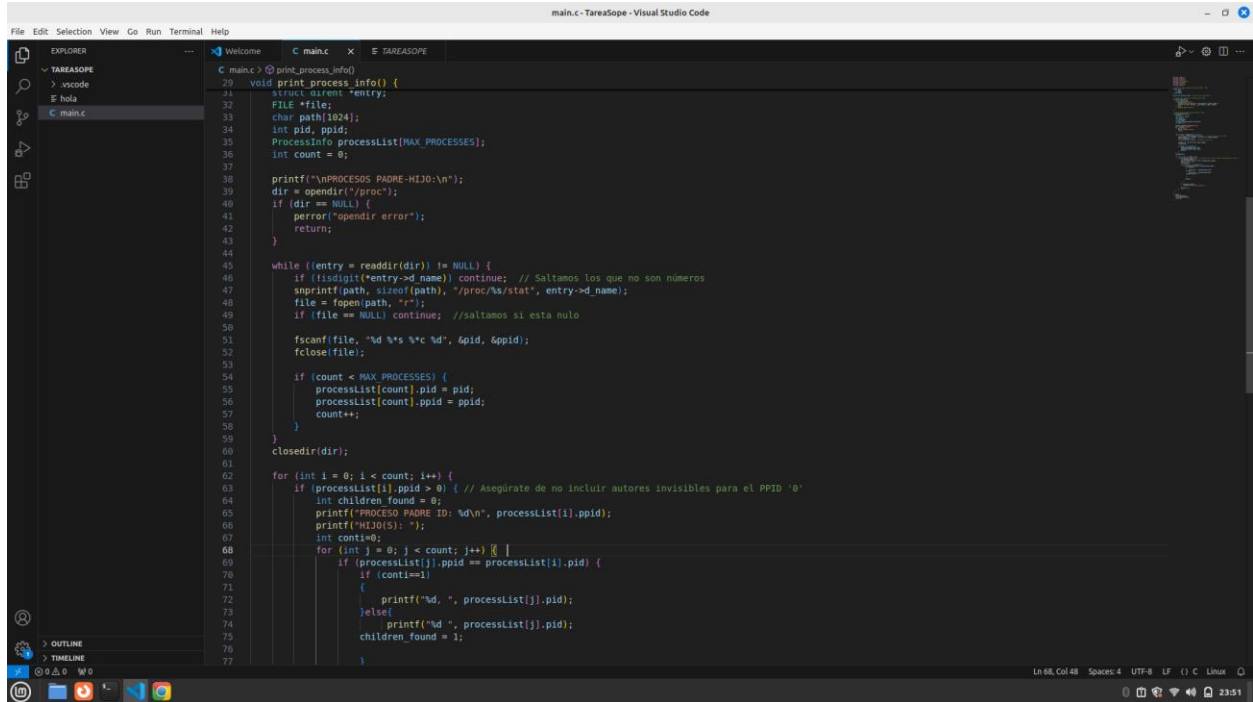
Programa



The screenshot shows the Visual Studio Code editor with a C program named `main.c`. The program is designed to print process information. It includes a `print_process_info()` function that iterates through a `processList` array, printing the PID of each process. The `main()` function calls `print_process_info()` and returns 0.

```
28 void print_process_info() {
29     printf("HIJO(S): ");
30     for (int j = 0; j < count; j++) {
31         if (processList[j].ppid == processList[i].pid) {
32             printf(" %d ", processList[j].pid);
33         }
34     }
35     printf("\n");
36 }
37
38 int main() {
39     print_ram_info();
40     print_process_info();
41     return 0;
42 }
```

The bottom status bar indicates the file is `main.c` at line 71, column 4, using UTF-8 encoding on a Linux system.



The screenshot shows the Visual Studio Code editor with a more complex C program named `main.c`. This program uses `opendir` and `readdir` to traverse the `/proc` directory, identifying parent and child processes. It prints the parent process ID and the child process IDs, along with the number of children found for each parent.

```
20 void print_process_info() {
21     struct dirent *entry;
22     FILE *file;
23     char path[1024];
24     int pid, ppid;
25     ProcessInfo processList[MAX_PROCESSES];
26     int count = 0;
27
28     printf("\nPROCESOS PADRE-HIJO:\n");
29     dir = opendir("/proc");
30     if (dir == NULL) {
31         perror("opendir error");
32         return;
33     }
34
35     while ((entry = readdir(dir)) != NULL) {
36         if (!isdigit(entry->d_name)) continue; // Saltamos los que no son números
37         sprintf(path, "/proc/%s/stat", entry->d_name);
38         file = fopen(path, "r");
39         if (file == NULL) continue; // saltamos si esta nulo
40
41         fscanf(file, "%d %s %c %d", &pid, &ppid);
42         fclose(file);
43
44         if (count < MAX_PROCESSES) {
45             processList[count].pid = pid;
46             processList[count].ppid = ppid;
47             count++;
48         }
49     }
50     closedir(dir);
51
52     for (int i = 0; i < count; i++) {
53         if (processList[i].ppid > 0) { // Asegúrate de no incluir autores invisibles para el PPID '0'
54             int children found = 0;
55             printf("PROCESO PADRE ID: %d\n", processList[i].ppid);
56             printf("HIJO(S): ");
57             int conti=0;
58             for (int j = 0; j < count; j++) {
59                 if (processList[j].ppid == processList[i].pid) {
60                     if (conti==1)
61                         printf(" %d ", processList[j].pid);
62                     else
63                         printf(" %d ", processList[j].pid);
64                     children found = 1;
65                 }
66             }
67             printf("\n");
68         }
69     }
70 }
```

The bottom status bar indicates the file is `main.c` at line 68, column 48, using UTF-8 encoding on a Linux system.