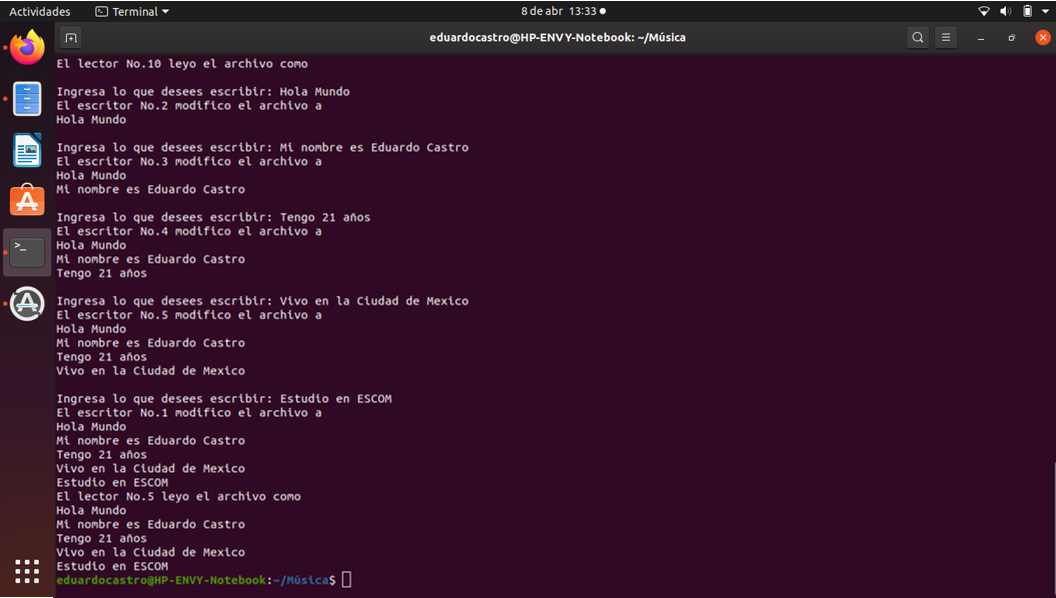
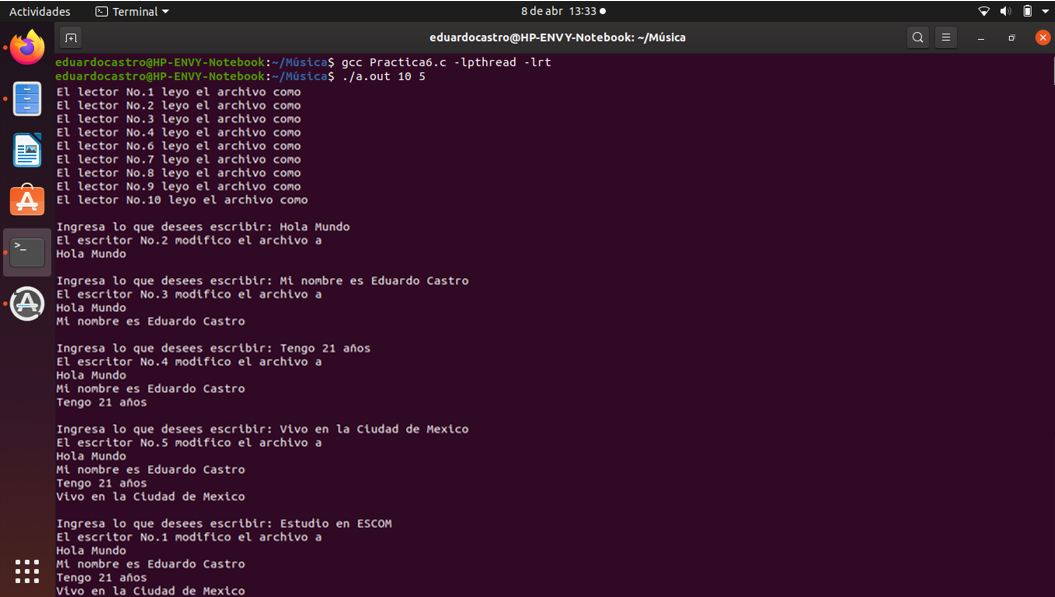
Código

1. #include <stdio.h>
2. #include <pthread.h>
3. #include <string.h>
4. #include <semaphore.h>
5. #include <stdlib.h>
7. **int** NumLeer=0;
8. sem\_t Escritor;
9. pthread\_mutex\_t mutex;
11. **void** \*FunEscritor(**void** \*wid){
12. **char** Cont, Mod[50];
14. **FILE**\* Archivo;
15. Archivo=fopen("Archivo.txt","a+");
17. **if**(Archivo==NULL)
18. perror("Error al intentar abrir el archivo");
19. sem\_wait(&Escritor);
20. printf("\nIngresa lo que desees escribir: ");
21. fgets(Mod, **sizeof**(Mod), stdin);
22. fprintf(Archivo, "%s", Mod);
23. printf("El escritor No.%d modifico el archivo a\n", \*(**int** \*)wid);
24. rewind(Archivo);
25. **while**((Cont=fgetc(Archivo)) != EOF)
26. printf("%c",Cont);
28. fclose(Archivo);
29. sem\_post(&Escritor);
30. }
32. **void** \*FunLector(**void** \*rid){
33. **FILE**\* Archivo;
34. **char** Cont;
36. **if**((Archivo=fopen("Archivo.txt", "r")) == NULL)
37. perror("Error al intentar abrir el archivo");
39. pthread\_mutex\_lock(&mutex);
40. NumLeer++;
41. **if**(NumLeer == 1)
42. sem\_wait(&Escritor);
44. pthread\_mutex\_unlock(&mutex);
46. printf("El lector No.%d leyo el archivo como\n",\*(**int** \*)rid);
47. **while**((Cont=fgetc(Archivo))!=EOF)
48. printf("%c", Cont);
49. pthread\_mutex\_lock(&mutex);
50. NumLeer--;
51. fclose(Archivo);
52. **if**(NumLeer == 0)
53. sem\_post(&Escritor);
54. pthread\_mutex\_unlock(&mutex);
55. }
57. **int** main(**int** argc, **char** **const** \*argv[]){
58. **FILE**\* Archivo;
59. Archivo=fopen("Archivo.txt", "w+");
60. fclose(Archivo);
62. **int** NumLectores, NumEscritores, i;
64. NumLectores=atoi(argv[1]);
65. NumEscritores=atoi(argv[2]);
67. pthread\_t Lectores[NumLectores], Escritores[NumEscritores];
68. pthread\_mutex\_init(&mutex, NULL);
69. sem\_init(&Escritor, 0, 1);
71. **int** Identificador[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
73. **for**(i=0; i<NumLectores; i++)
74. pthread\_create(&Lectores[i], NULL, (**void** \*)FunLector, (**void** \*)&Identificador[i]);
76. **for**(i=0; i<NumEscritores; i++)
77. pthread\_create(&Escritores[i], NULL, (**void** \*)FunEscritor, (**void** \*)&Identificador[i]);
79. **for**(i=0; i<NumLectores; i++)
80. pthread\_join(Lectores[i], NULL);
82. **for**(i=0; i<NumEscritores; i++)
83. pthread\_join(Escritores[i], NULL);
85. pthread\_mutex\_destroy(&mutex);
86. sem\_destroy(&Escritor);
87. **return** 0;
88. }

Capturas de pantalla



Explicación

Creamos un archivo llamado Archivo.txt para almacenar lo que escriben los escritores.

Almacenamos los parámetros que ingreso el usuario.

Mediante un for creamos tantos hilos lectores como el usuario haya ingresado.

Verificamos si es posible abrir el archivo.

Bloqueamos al escritor mientras haya un lector.

Imprimimos lo que el lector leyó.

Cerramos el archivo.

Si es el último lector este despertará al escritor.

Desbloqueamos al mutex.

Mediante un for creamos tantos hilos escritores como el usuario haya ingresado.

Declaramos un arreglo de caracteres para almacenar lo que el escritor escriba.

Abrimos el archivo que previamente habíamos creado.

Solicita que ingresemos el contenido a almancenar.

Almacenamos lo que el usuario escribió dentro de Archivo.txt.

Imprimimos el identificador del escritor que modificó el archivo.

Imprimimos el contenido del archivo.

Cerramos Archivo.txt.

Mediante un for unimos tantos hilos lectores como el usuario haya ingresado.

Mediante un for unimos tantos hilos escritores como el usuario haya ingresado.

Destruimos el hilo mutex.

Destruimos sem escritor.