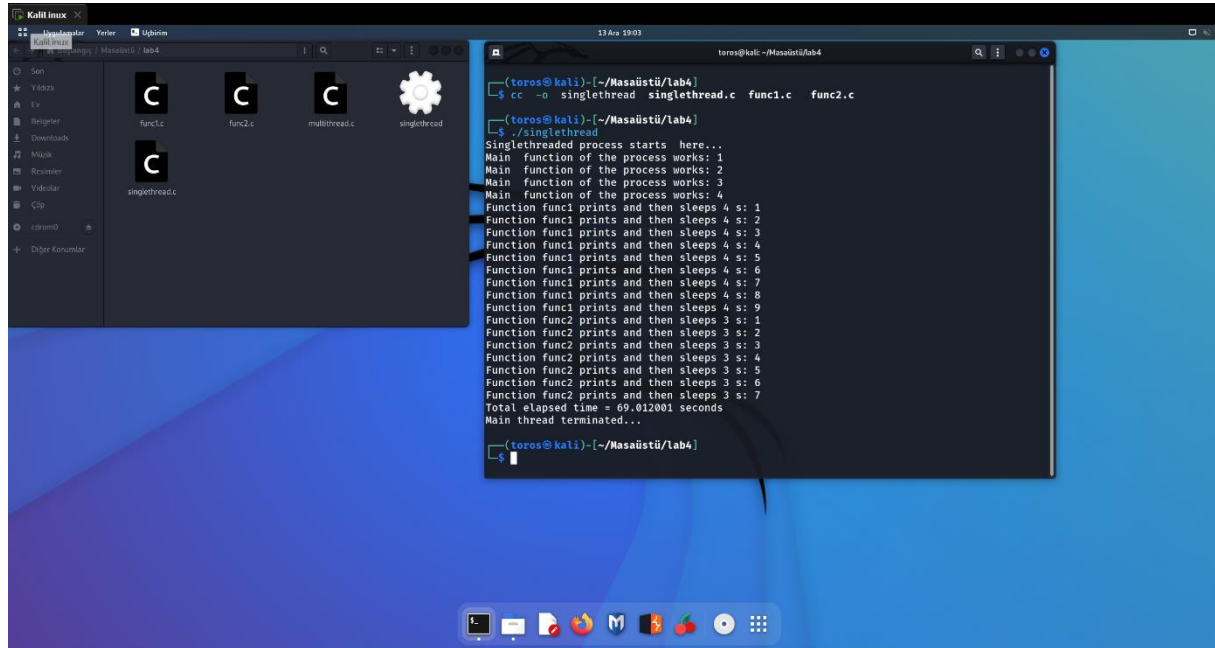


1-2-3)

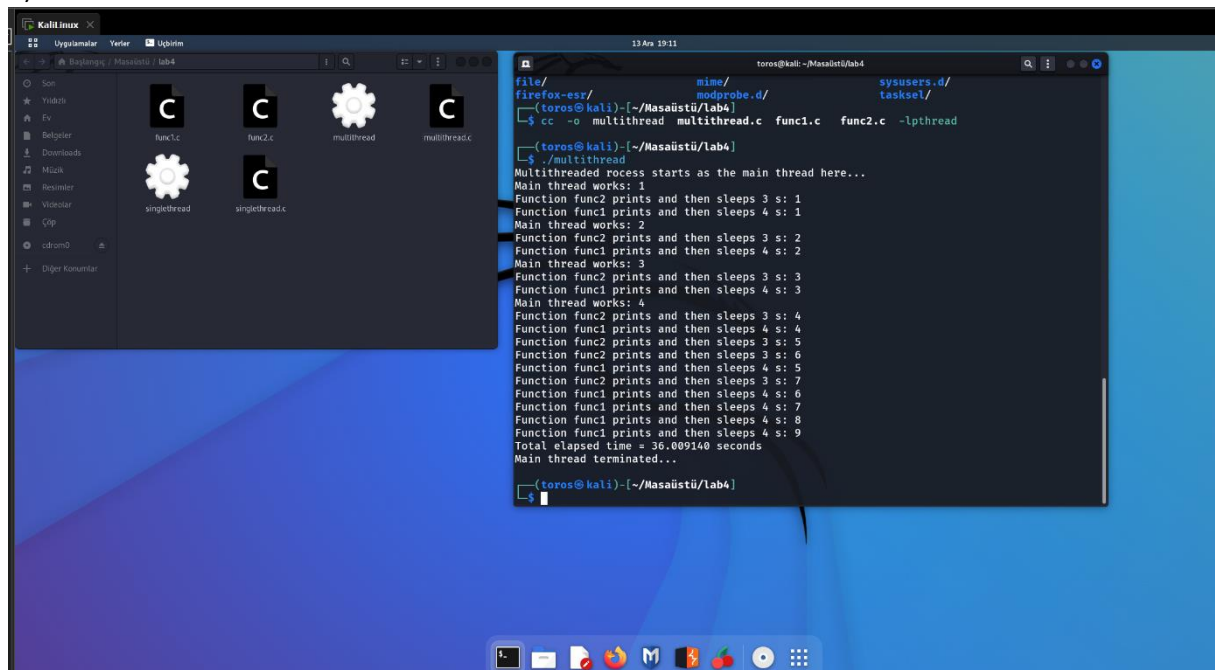
Lab3a adında klasör oluşturdum. Buraya kod dosyalarını attım



```
(toros@kali) ~/Masaüstü/lab4
$ cc -o singlethread singlethread.c func1.c func2.c
(toros@kali) ~/Masaüstü/lab4
$ ./singlethread
Singlethreaded process starts here...
Main function of the process works: 1
Main function of the process works: 2
Main function of the process works: 3
Main function of the process works: 4
Function func1 prints and then sleeps 4 s: 1
Function func1 prints and then sleeps 4 s: 2
Function func1 prints and then sleeps 4 s: 3
Function func1 prints and then sleeps 4 s: 4
Function func1 prints and then sleeps 4 s: 5
Function func1 prints and then sleeps 4 s: 6
Function func1 prints and then sleeps 4 s: 7
Function func1 prints and then sleeps 4 s: 8
Function func1 prints and then sleeps 4 s: 9
Function func2 prints and then sleeps 3 s: 1
Function func2 prints and then sleeps 3 s: 2
Function func2 prints and then sleeps 3 s: 3
Function func2 prints and then sleeps 3 s: 4
Function func2 prints and then sleeps 3 s: 5
Function func2 prints and then sleeps 3 s: 6
Function func2 prints and then sleeps 3 s: 7
Total elapsed time = 69.012001 seconds
Main thread terminated...
(toros@kali) ~/Masaüstü/lab4
$
```

Main fonksiyon 3 er saniyelik 4 çevrim yaptı. Ardından Func1 ve Func2 metodları sıra ile çağrıldı. Birinci fonksiyon 4 er saniyelik ara ile ekrana çıktı verdi, 9 çevrim ile toplam 36 saniyelik işlem yapıldı. İkinci fonksiyon 3 er saniyelik ara ile ekrana çıktı verdi 7 çevrim ile toplam 21 saniyelik gecikme olması bekleniyor(toplam 69). Program başlangıç saati ile bitiş saatinin farkını aldı ve toplam geçen süre 69.01s oldu.

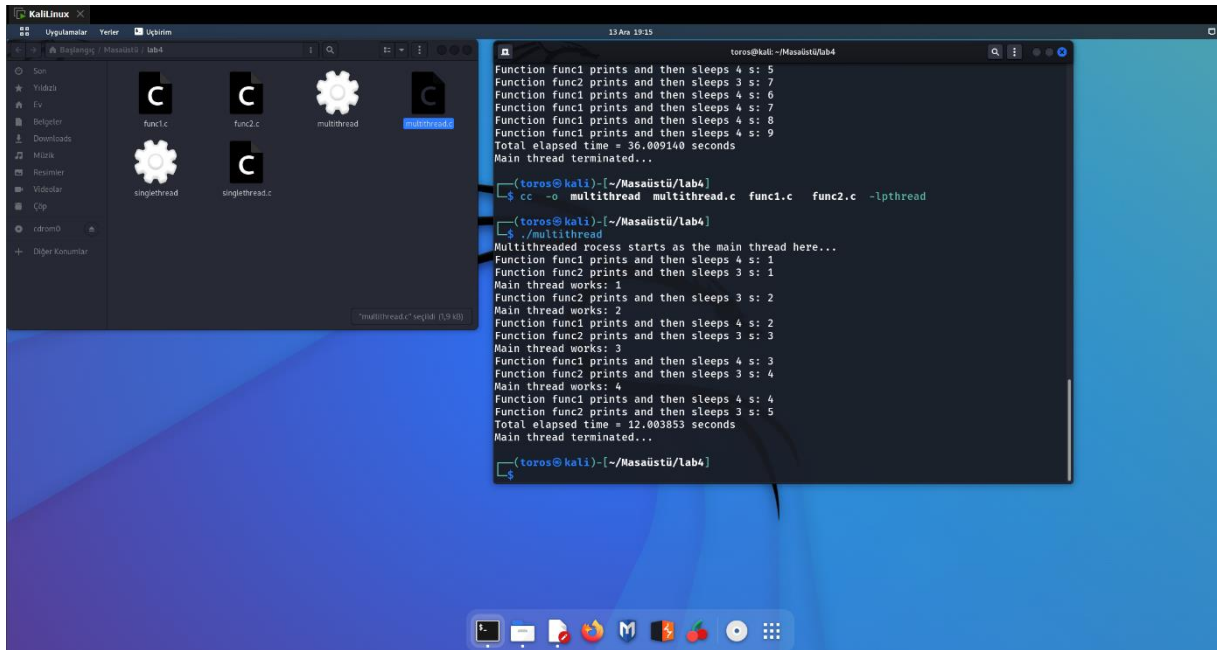
4)



```
file/      mime/      sysusers.d/
firefox-esr/  mndprobe.d/  tasksel/
(toros@kali) ~/Masaüstü/lab4
$ cc -o multithread multithread.c func1.c func2.c -lpthread
(toros@kali) ~/Masaüstü/lab4
$ ./multithread
Multithreaded process starts as the main thread here...
Main thread works: 1
Function func2 prints and then sleeps 3 s: 1
Function func1 prints and then sleeps 4 s: 1
Main thread works: 2
Function func2 prints and then sleeps 3 s: 2
Function func1 prints and then sleeps 4 s: 2
Main thread works: 3
Function func2 prints and then sleeps 3 s: 3
Function func1 prints and then sleeps 4 s: 3
Main thread works: 4
Function func2 prints and then sleeps 3 s: 4
Function func1 prints and then sleeps 4 s: 4
Function func2 prints and then sleeps 3 s: 5
Function func1 prints and then sleeps 4 s: 5
Function func2 prints and then sleeps 3 s: 6
Function func1 prints and then sleeps 4 s: 6
Function func2 prints and then sleeps 3 s: 7
Function func1 prints and then sleeps 4 s: 7
Function func1 prints and then sleeps 4 s: 8
Function func1 prints and then sleeps 4 s: 9
Total elapsed time = 36.009140 seconds
Main thread terminated...
(toros@kali) ~/Masaüstü/lab4
$
```

Singlethread işlem açısından bakınca yukarıdaki ekranda süreler fun2 gecikme süresini verince 36 saniye olduğu anlaşılıyor.

5)



```
Function func1 prints and then sleeps 4 s: 5
Function func2 prints and then sleeps 3 s: 7
Function func1 prints and then sleeps 4 s: 6
Function func1 prints and then sleeps 4 s: 7
Function func1 prints and then sleeps 4 s: 8
Function func1 prints and then sleeps 4 s: 9
Total elapsed time = 36.009140 seconds
Main thread terminated...

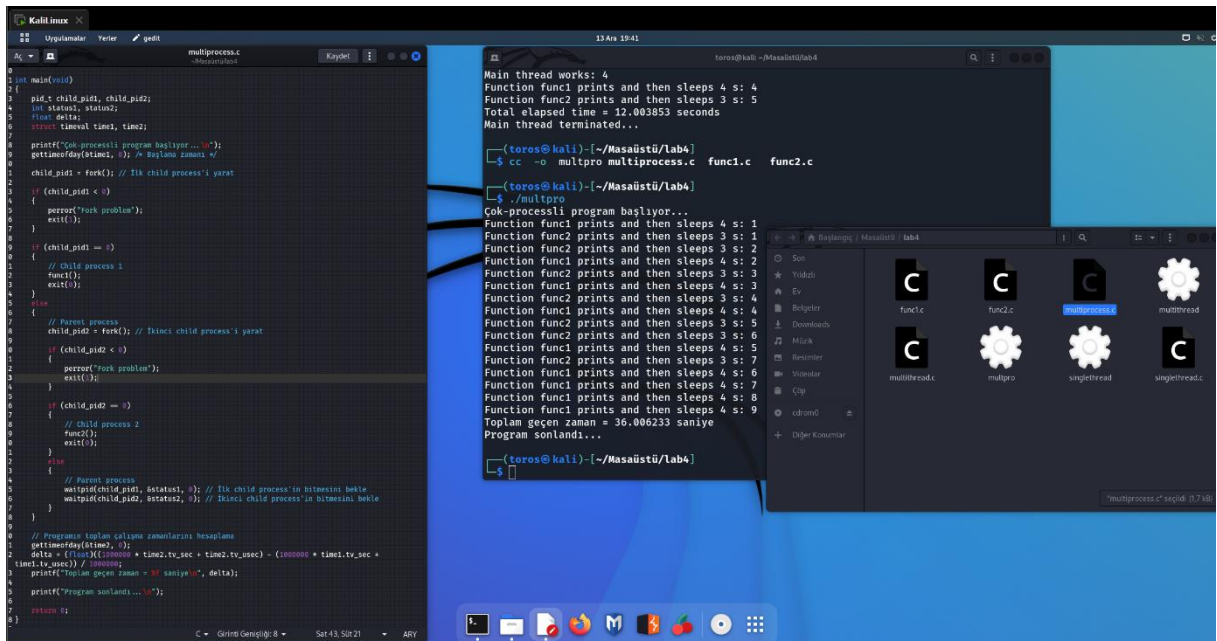
(toros@kali) ~/Masaüstü/Lab4
$ cc -o multithread multithread.c func1.c func2.c -lpthread

(toros@kali) ~/Masaüstü/Lab4
$ ./multithread
Multithreaded process starts as the main thread here...
Function func1 prints and then sleeps 4 s: 1
Function func2 prints and then sleeps 3 s: 1
Main thread works: 1
Function func2 prints and then sleeps 3 s: 2
Main thread works: 2
Function func1 prints and then sleeps 4 s: 2
Function func2 prints and then sleeps 3 s: 3
Main thread works: 3
Function func1 prints and then sleeps 4 s: 3
Function func2 prints and then sleeps 3 s: 4
Main thread works: 4
Function func1 prints and then sleeps 4 s: 4
Function func2 prints and then sleeps 3 s: 5
Total elapsed time = 12.003853 seconds
Main thread terminated...

(toros@kali) ~/Masaüstü/Lab4
$
```

pthread_joiner kaldırılınca sonlanması beklenen thread sonlanana kadar beklenmiyor bu yüzden zaman kıaldı. Süre 12 saniye yani 3 te 1 oranda azalmış

6)



```
Main thread works: 4
Function func1 prints and then sleeps 4 s: 4
Function func2 prints and then sleeps 3 s: 5
Total elapsed time = 12.003853 seconds
Main thread terminated...

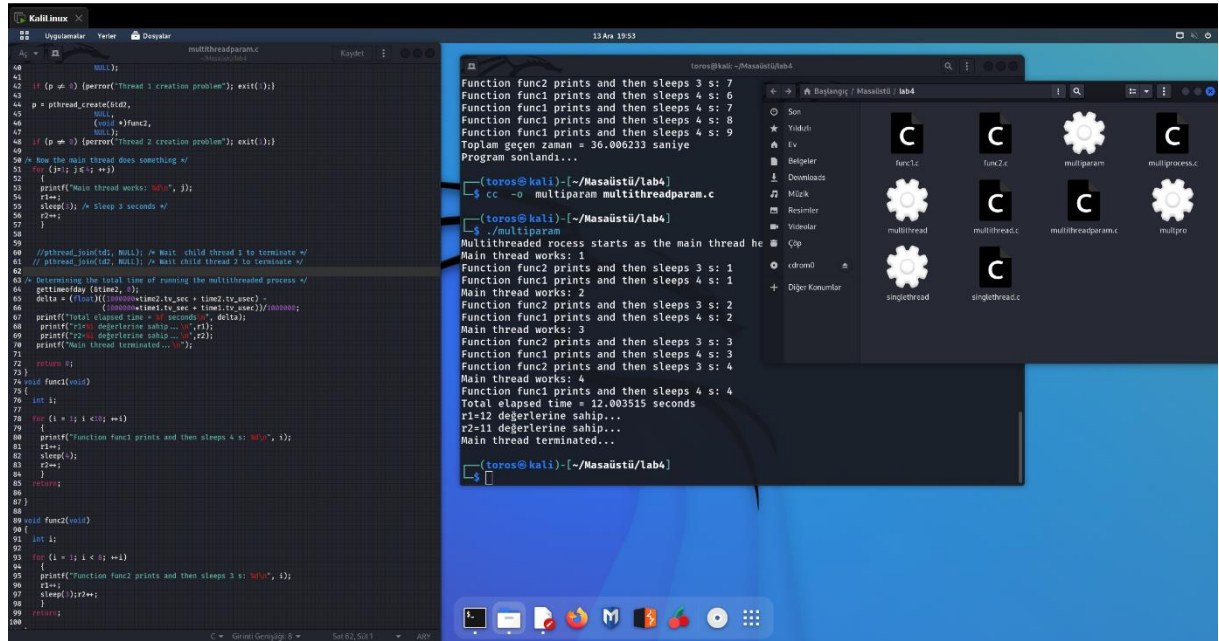
(toros@kali) ~/Masaüstü/Lab4
$ cc -o multipro multiprocess.c func1.c func2.c

(toros@kali) ~/Masaüstü/Lab4
$ ./multipro
Çok-processli program başlıyor...
Function func1 prints and then sleeps 4 s: 1
Function func2 prints and then sleeps 3 s: 1
Function func1 prints and then sleeps 4 s: 2
Function func2 prints and then sleeps 3 s: 2
Function func1 prints and then sleeps 4 s: 3
Function func2 prints and then sleeps 3 s: 3
Function func1 prints and then sleeps 4 s: 4
Function func2 prints and then sleeps 3 s: 4
Function func1 prints and then sleeps 4 s: 5
Function func2 prints and then sleeps 3 s: 5
Function func1 prints and then sleeps 4 s: 6
Function func2 prints and then sleeps 3 s: 6
Function func1 prints and then sleeps 4 s: 7
Function func2 prints and then sleeps 3 s: 7
Function func1 prints and then sleeps 4 s: 8
Function func2 prints and then sleeps 3 s: 8
Function func1 prints and then sleeps 4 s: 9
Toplam geçen zaman = 36.006233 saniye
Program sonlandı...

(toros@kali) ~/Masaüstü/Lab4
$
```

çoklu işlem kullanılarak iki ayrı sürecin eşzamanlı olarak çalıştırılmasının ardından toplam geçen zaman 36 s olarak ekrana yazdırılmış

7)



The screenshot shows a Kali Linux desktop environment. On the left, a terminal window displays the source code of a C program named `multithreadparam.c`. The code defines two functions, `func1` and `func2`, which print messages and sleep for 4 seconds. The `main` function creates two child threads, `func1` and `func2`, and then sleeps for 12 seconds. On the right, a file manager window shows the contents of the `~/Masaüstü/Lab4` directory, which includes files like `func1.c`, `func2.c`, `multiparam`, `multithread`, `multithreadparam.c`, and `multithread.c`. Below the file manager, a terminal window shows the output of the program. It displays the execution of `cc -o multiparam multithreadparam.c` and the execution of `./multiparam`. The output shows the main thread starting, followed by the execution of `func1` and `func2`, which print their respective messages and sleep for 4 seconds. The main thread then prints the total elapsed time, which is 12.003515 seconds, and terminates.

Main thread, bir döngü içinde belirli bir süre boyunca işlemlerini gerçekleştirirken, child thread'ler da kendi içlerinde belirli işlemleri gerçekleştirir ve ortak global değişkenlere (**r1** ve **r2**) erişim sağlar. **sleep** fonksiyonları, işlemlerin belirli süreler boyunca durmasını simüle eder. Programın toplam çalışma süresi, başlama ve bitiş zamanları arasındaki fark olarak hesaplanır.

R1 = 12

R2 = 11 olarak yazdırıldı.