



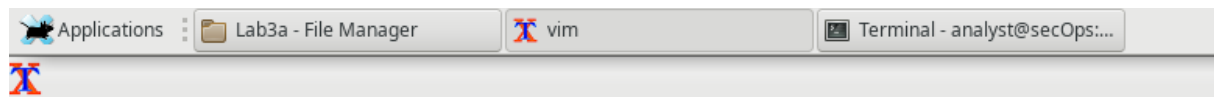
**MERSİN UNIVERSITY**

**COMPUTER ENGINEERING DEPARTMENT**

**Sistem Programlama**

**2022/2023 Fall Semester**

<b>Assignment Date:</b>	23/11/2022
<b>Assignment Name:</b>	Lab4 Ödev
<b>Student Name:</b>	İlhan DAŞDEMİR
<b>Student ID:</b>	22220570005



```
int r1 = 0, r2 = 0; /* Global variables, accessible to functions if necessary */

int
main(void)
{
    pthread_t    td1, td2; /* Descriptors for two child threads, not used here */
    int p; /* Not used here */
    int j;
    float delta;
    struct timeval time1, time2;

    printf("Singlethreaded process starts here...\n");
    gettimeofday(&time1, 0); /* Starting time */

    /* Creation of two child threads
    p = pthread_create(&td1,
        NULL,
        (void *)func1,
        NULL);

    if (p != 0) {perror("Thread 1 creation problem"); exit(1);}

    p = pthread_create(&td2,
        NULL,
        (void *)func2,
        NULL);
    if (p != 0) {perror("Thread 2 creation problem"); exit(1);}
    */

    /* Now the process does something by its main thread */
    for (j=1; j<=4; ++j)
    {
        printf("Main function of the process works: %d\n", j);
        sleep(3); /* Sleep 3 seconds */
    }

    /* Then the process calls sequentially two functions */
    func1();
    func2();

    /* Determining the total time of running the process */
    gettimeofday (&time2, 0);
    delta = (float)((1000000*time2.tv_sec + time2.tv_usec) -
        (1000000*time1.tv_sec + time1.tv_usec))/1000000;
    printf("Total elapsed time = %f seconds\n", delta);

    printf("Main thread terminated...\n");

    return 0;
}

void func1(void)
{
    int i;
    for(i=1;i<10;i++)
    {
        printf("Func1 yazari ve 4 sn uyu:%d\n",i);
        sleep(4);
    }
    return;
}

void func2(void)
{
    int i;
    for(i=1;i<10;i++){
        printf("Func2 3sn uyur:%d\n",i);
        sleep(3);
    }return;
}
}
```

Singlethread, Func1 ve Func2 birleşik hali

### Deneyler:

- 1- Linux home/analyst klasörü altında Lab3a adında klasör oluşturdum. Buraya singlethread.c, multithread.c, Func1.c ve Func2.c dosyalarını attım.
- 2- **cc -o singlethread singlethread.c func1.c func2.c** komutunu kullanarak 3 ayrı kod dosyasını tek singlethread.c dosyası altında birleştirmeye çalıştım ancak sebebini bulamadığım bir hata oluştu. Singlethread.c dosyasını kaybettim. Bunun üzerine Vim ile singlethread dosyasını açıp fonksiyon içeriklerini elle dosyanın içine yazdım.
- 3- Terminalde **cc singlethread.c** komutu ile kodu derleyerek a.out adlı dosyayı elde ettim. **./a.out** komutu ile dosyayı çalıştırdım ve aşağıdakini elde ettim

```
[root@secOps Lab3a]# cc -o singlethread.c func1.c func2.c^C
[root@secOps Lab3a]# ./a.out
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
Func1 yazar ve 4 sn uyu:1
rFunc1 yazar ve 4 sn uyu:2
rFunc1 yazar ve 4 sn uyu:3
rFunc1 yazar ve 4 sn uyu:4
rFunc1 yazar ve 4 sn uyu:5
rFunc1 yazar ve 4 sn uyu:6
rFunc1 yazar ve 4 sn uyu:7
rFunc1 yazar ve 4 sn uyu:8
rFunc1 yazar ve 4 sn uyu:9
rFunc2 3sn uyur:1
Func2 3sn uyur:2
Func2 3sn uyur:3
Func2 3sn uyur:4
Func2 3sn uyur:5
Func2 3sn uyur:6
Func2 3sn uyur:7
Func2 3sn uyur:8
Func2 3sn uyur:9
Total elapsed time = 75.005615 seconds
Main thread terminated...
[root@secOps Lab3a]#
```

Ana 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, 9 çevrim ile toplam 27 saniyelik gecikme olması bekleniyor(toplam 75). Program başlangıç saati ile bitiş saatinin farkını aldı ve toplam geçen süre 75,005615sn oldu.

4. multithread.c dosyasını compile ederken kullanılacak kütüphaneyi aşağıdaki şekilde değiştirince kod çalıştı ve ekran görüntüsü aşağıdadır

cc multithread.c /usr/lib/libpthread.so

```
[root@secOps Lab3a]# cc multithread.c /usr/lib/libpthread.so
[root@secOps Lab3a]# ls
a.out func1.c func2.c lab4.pdf mt.c multithread.c
[root@secOps Lab3a]# ./a.out
Multithreaded rocess 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 func2 prints and then sleeps 3 s: 6
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 = 36.003345 seconds
Main thread terminated...
[root@secOps Lab3a]#
```

Singlethread işlem açısından bakınca yukarıdaki ekranda süreler toplanırsa geçen sürenin 57 saniye olması gerekiyor. Oysaki çoklu işlem ile iki child işlem için beklenen gecikme 4+3 saniye olması gerekirken, sanki iki işlem paralel yürümüş ve süre büyük işlem süresi olan 4 saniye sürmüş gibi görünüyor. 3 saniyelik işlem 7 kez çalışmış bu 21 saniye ediyor. 57-21=36 saniye elde edilir. Bu da çoklu işlemin boşuna beklemeyi önlediğini österiyor.

5. pthread\_joinler kaldırılınca sonlanması beklenen thread sonlanana kadar beklenmiyor bu yüzden zaman kısaldı. Süre 12 saniye yani 3 te 1 oranda azalmış

```
ilhan@linux-mint: ~/lab4
Dosya Düzenle Görünüm Ara Uçbirim Yardım
ilhan@linux-mint:~/lab4$ make clean
rm a.out
ilhan@linux-mint:~/lab4$ ls
func1.c func2.c lab4.pdf makefile multithread.c singlethread.c
ilhan@linux-mint:~/lab4$ make
cc multithread.c
ilhan@linux-mint:~/lab4$ ./a.out
Multithreaded rocess starts as the main thread here...
Function func1 prints and then sleeps 4 s: 1
Main thread works: 1
Function func2 prints and then sleeps 3 s: 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 = 12.000424 seconds
Main thread terminated...
ilhan@linux-mint:~/lab4$
```

6.

```
tunc1();

//pthread_join(td1, NULL); /* Wait child thread 1 to terminate */
// pthread_join(td2, NULL); /* Wait child thread 2 to terminate */

/* Determining the total time of run */
gettimeofday(&time2, 0);
delta = (float)((1000000*time2.tv_sec +
(1000000*time1.tv_sec) +
(1000000*time1.tv_usec) -
(1000000*time2.tv_sec +
(1000000*time2.tv_usec) +
(1000000*time2.tv_usec)));
printf("Total elapsed time = %f\n", delta);

printf("Main thread terminated...\n");

return 0;
}

void func1(void)
{
    int i;

    for (i = 1; i <10; ++i)
    {
        printf("Main thread prints and then sleeps 4 s: %d\n", i);
        sleep(4);
    }
}

void func2(void)
{
    int i;

    for (i = 1; i < 8; ++i)
    {
        printf("Thread prints and then sleeps 3 s: %d\n", i);
        sleep(3);
    }
}
```

Program yanlış olabilir. Ekran görüntüsü yukarıdaki şekilde oldu.

7. r1 ve r2 değişkenleri kod her çalıştığında farklı değerler aldı.

```
ilhan@linux-mint: ~/lab4
Dosya  Düzenle  Görünüm  Ara  Uçbirim  Yardım
ilhan@linux-mint:~/lab4$ cc multithreadparam.c
ilhan@linux-mint:~/lab4$ ls
a.out  func1.c  func2.c  lab4.pdf  makefile  multiprocess.c  multithread.c  multithreadparam.c
ilhan@linux-mint:~/lab4$ ./a.out
Multithreaded process starts as the main thread here...
Function func1 prints and then sleeps 4 s: 1
Main thread works: 1
Function func2 prints and then sleeps 3 s: 1
Main thread works: 2
Function func2 prints and then sleeps 3 s: 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
Main thread works: 4
Function func2 prints and then sleeps 3 s: 4
Function func1 prints and then sleeps 4 s: 4
Total elapsed time = 12.000395 seconds
r1=12 değerlerine sahip...
r2=10 değerlerine sahip...
Main thread terminated...
ilhan@linux-mint:~/lab4$

ilhan@linux-mint: ~/lab4
Dosya  Düzenle  Görünüm  Ara  Uçbirim  Yardım
ilhan@linux-mint:~/lab4$ make clean
rm a.out
ilhan@linux-mint:~/lab4$ cc multithreadparam.c
ilhan@linux-mint:~/lab4$ ./a.out
Multithreaded process starts as the main thread here...
Function func1 prints and then sleeps 4 s: 1
Main thread works: 1
Function func2 prints and then sleeps 3 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
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 = 12.000477 seconds
r1=13 değerlerine sahip...
r2=11 değerlerine sahip...
Main thread terminated...
ilhan@linux-mint:~/lab4$

ilhan@linux-mint: ~/lab4
Dosya  Düzenle  Görünüm  Ara  Uçbirim  Yardım
ilhan@linux-mint:~/lab4$ ls
a.out  func1.c  func2.c  lab4.pdf  makefile  multiprocess.c  multithread.c  multithreadparam.c
ilhan@linux-mint:~/lab4$ ./a.out
Multithreaded process starts as the main thread here...
Function func1 prints and then sleeps 4 s: 1
Main thread works: 1
Function func2 prints and then sleeps 3 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
Function func2 prints and then sleeps 3 s: 4
Main thread works: 4
Function func2 prints and then sleeps 3 s: 4
Function func1 prints and then sleeps 4 s: 4
Total elapsed time = 12.000441 seconds
r1=12 değerlerine sahip...
r2=11 değerlerine sahip...
Main thread terminated...
ilhan@linux-mint:~/lab4$
```