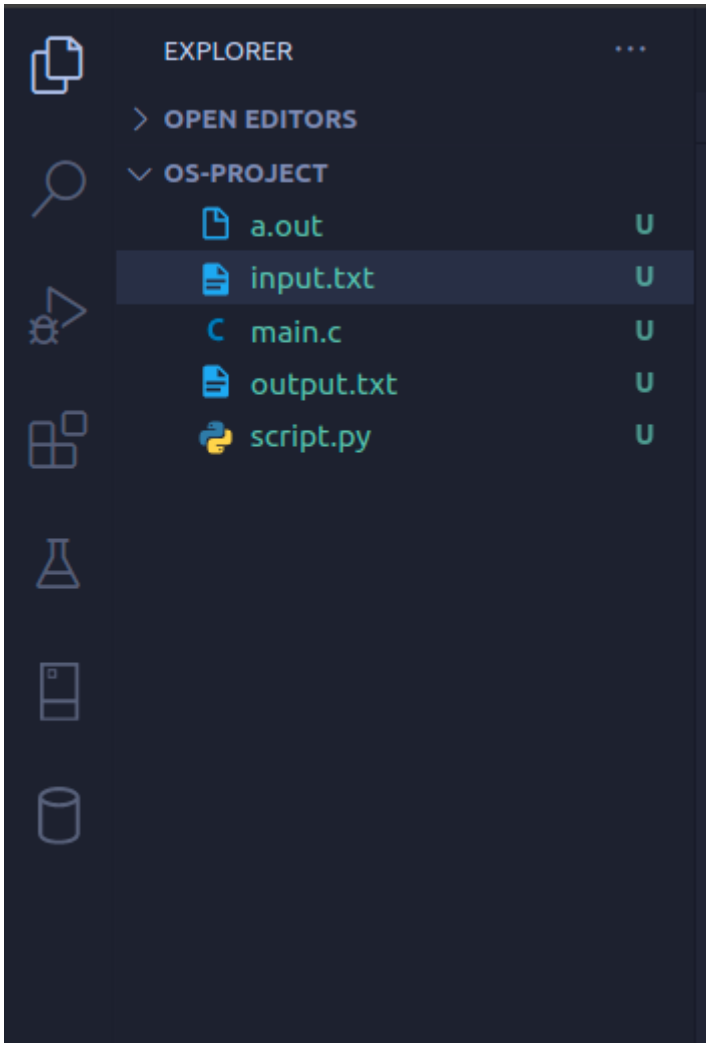
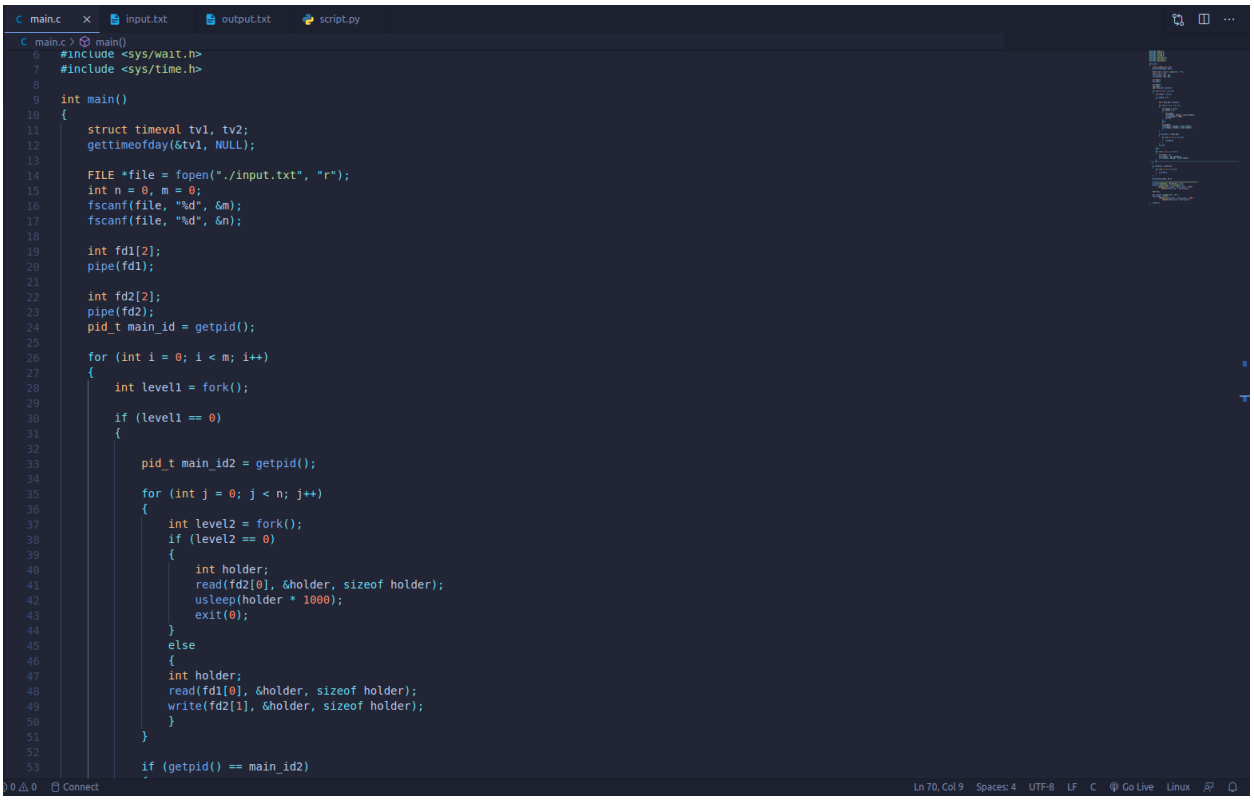
پروژه سیستم عامل
دکتر خانمیرزا

علی الهی

توضیحات فایل های موجود در پروژه



فایل اصلی برنامه :



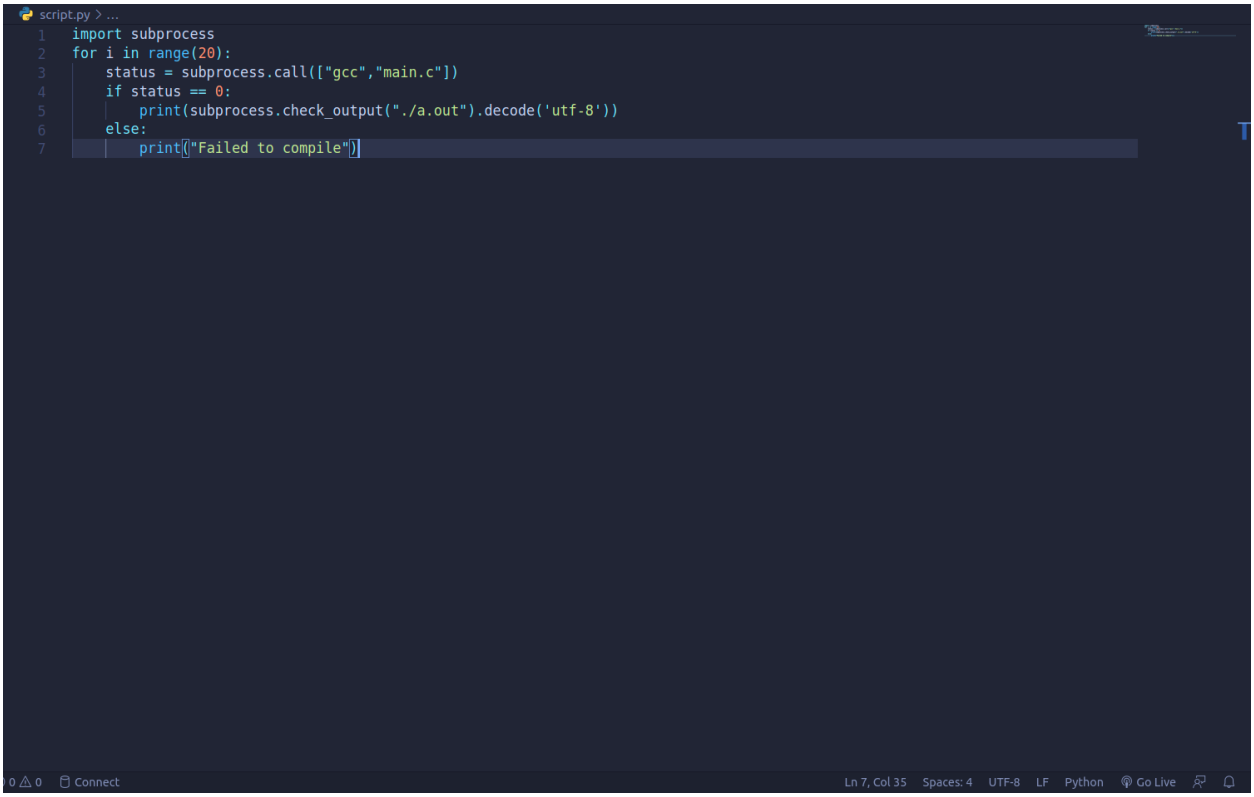
```c
#include <sys/wait.h>
#include <sys/time.h>

int main()
{
    struct timeval tv1, tv2;
    gettimeofday(&tv1, NULL);

    FILE *file = fopen("./input.txt", "r");
    int n = 0, m = 0;
    fscanf(file, "%d", &m);
    fscanf(file, "%d", &n);

    int fd1[2];
    pipe(fd1);

    int fd2[2];
    pipe(fd2);
    pid_t main_id = getpid();

    for (int i = 0; i < m; i++)
    {
        int level1 = fork();

        if (level1 == 0)
        {

            pid_t main_id2 = getpid();

            for (int j = 0; j < n; j++)
            {
                int level2 = fork();
                if (level2 == 0)
                {
                    int holder;
                    read(fd2[0], &holder, sizeof holder);
                    usleep(holder * 1000);
                    exit(0);
                }
                else
                {
                int holder;
                read(fd1[0], &holder, sizeof holder);
                write(fd2[1], &holder, sizeof holder);
                }
            }

            if (getpid() == main_id2)
```

```c
            if (getpid() == main_id2)
            {
                for (int j = 0; j < n; j++)
                {
                    wait(NULL);
                }
            }
            exit(0);
        }
        else
        {
        for (int j = 0; j < n; j++)
        {
            int holder = 0;
            fscanf(file, "%d", &holder);
            write(fd1[1], &holder, sizeof holder);
        }
        }
    }

    if (getpid() == main_id)
    {
        for (int i = 0; i < m; i++)
        {
            wait(NULL);
        }
    }

    gettimeofday(&tv2, NULL);

    printf("================================================");
    printf("\nConfigue : %d and %d", m, n);
    printf("\nTotal time = %f seconds\n \n",
           (double)(tv2.tv_usec - tv1.tv_usec) / 1000 +
               (double)(tv2.tv_sec - tv1.tv_sec));

    FILE *fp;

    fp = fopen("./output.txt", "a+");
    fprintf(fp, "%f,\n",
            (double)(tv2.tv_usec - tv1.tv_usec) / 1000 +
                (double)(tv2.tv_sec - tv1.tv_sec));

    return 0;
}
```

فایل input که در آن کانفیگ برنامه و sleep Time های هرکدام وجود دارد :

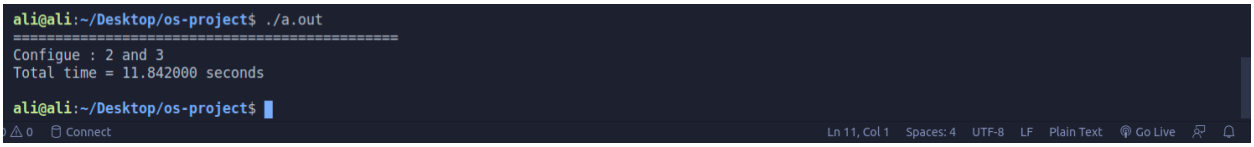
فایل اسکریپت پایتون :

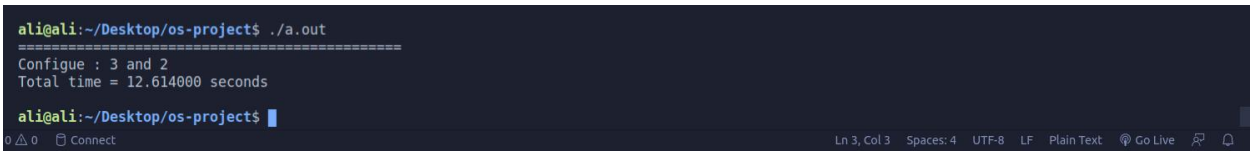
وظیفه این فایل اجرای کامند های gcc main.c و ./a.out. برای 20 بار پشت سر هم است تا بتوان میانگین زمان ها را حساب کرد.



فایل **output.txt** که زمان اجرا را ذخیره میکند :

مثال 1 :





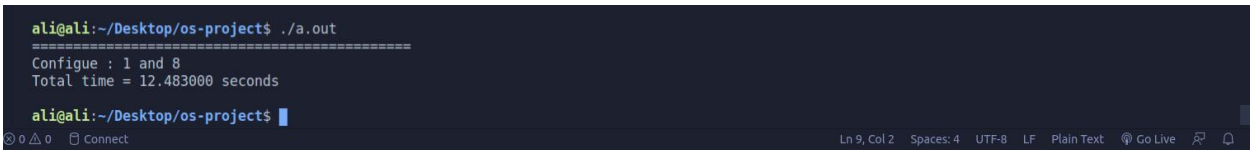
همانطور که مشخص است زمان اجرای کانفیگ ها به ترتیب :

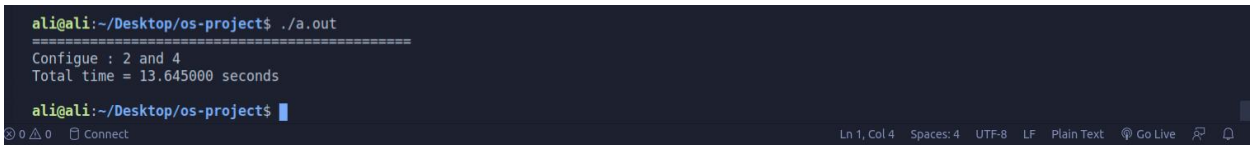
$$[1,6] < [2,3] < [3,2] < [6,1]$$

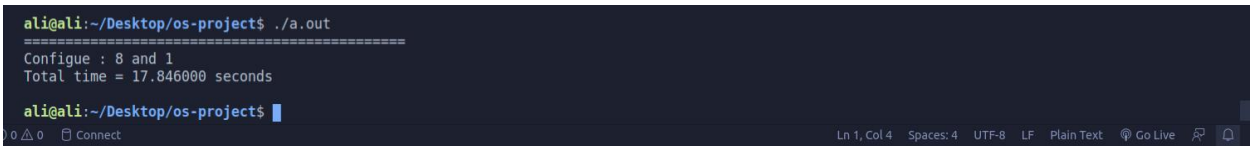
است.

هرچه level اول عدد کوچکتری داشته باشد زمان اجرا نیز کمتر خواهد بود.

مثال 2 :







همانطور که مشخص است زمان اجرای کانفیگ ها به ترتیب :

$$[1,8] < [2,4] < [4,2] < [8,1]$$

است.

مانند مثال قبل هرچه level اول عدد کوچکتری داشته باشد زمان اجرا نیز کمتر خواهد بود.

نکته:

زمان های اجرا به مشخصات فنی سیستمی که برنامه روی آن اجرا میشود بستگی دارد.

و نزدیک بودن زمان های اجرای کانفیگ های مختلف به علت تعداد core های لپتاپ میباشد.

نمودار مثال اول :

نمودار مثال دوم :