

## بخش 1)

نوشتن روی sharde memory:

ابتدا هدر های لازم را فراخوانی میکنیم

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <sys/stat.h>
```

سپس متغیر های لازم را تعریف میکنیم

```
int segment_id; // این متغیر شماره و شناسه حافظه اشتراکی را مشخص میکند
char* shared_memory; // یک پوینتر به حافظه اشتراکی
int shared_segment_size = 100;
```

**1 استراکچر مربوط به یافر shmids**

```
int segment_size;
```

سپس یک مقدار یونیک را با تابع ftok() برای ورودی دادن به shmget ایجاد میکنیم

```
Key_t key = ftok("shmfile",65);
```

تابع ftok یک کلید را بر اساس path و id برمی گرداند. این تابع برای همه مسیرهایی که با همان شناسه فراخوانی می شوند به همان پرونده اشاره می کنند کلید یکسانی را برمی گرداند. اگر ftok() با مقادیر ID مختلف یا مسیر به فایل های مختلف در یک سیستم پرونده یکسان باشد ، کلیدهای مختلف را برمی گرداند.

سپس با استفاده از تابع زیر یک حافظه اشتراکی را ایجاد کرده (اگر از قبل نباشد) و در پارامتر اول key و در دوم سایز حافظه و در سومین در پارامتر سوم نوع دسترسی را مشخص کرده و ای دی مورد نظر را در خروجی میریزیم

```
segment_id = shmget(IPC_PRIVATE, shared_segment_size, S_IRUSR | S_IWUSR);
```

سپس باید ان حافظه را به فضای ادرسی خود اضافه کنیم که تن نیز با تابع shmat ضرورت میگیرد که پارامتر اول ان identifier حافظه مورد نظر است و دومی ادرس حافظه که ما ان را null گذاشتیم و مشخص نکردیم و سومین پارامتر نوع دسترسی به ان است

```
shared_memory = (char*) shmat (segment_id, NULL, 0);
```

سپس با استفاده از تبع اطلاعاتی shmctl اطلاعات مربوط ی ان حافظه را میگیریم که ورودی اول ان همان شناسه حافظه اشتراکی و پارامتر دوم وضعیت ان و در سومین پارامتر هم داده ساختار بافر را میگذاریم

```
shmctl(segment_id, IPC_STAT, &shmbuffer);
```

سپس با تابع `sprintf` که اولین ورودی آن حافظه اشتراکی است اطلاعات را در `shared` memory `shmctl(segment_id, IPC_STAT, &shmbuffer);`

```
sprintf(shared_memory, "%s", argv[1]);
```

سپس حافظه اشتراکی را دیتج میکنیم  
`shmdt(shared_memory);`

برنامه خواندن از `shared memory`:

این برنامه عین برنامه بالاست و دبرای دسترسی به `shared memory` همان `key` را میسازیم و به آن می دهیم و برای دیدن اطلاعات داخل `shared memory` آن را به صورت پایین چاپ میکنیم

`printf("read from shared memory : %s", shared_memory)` که `shared memory` همان ادرس حافظه اشتراکی است



