

22K-  
+4373

22K-4373

Date:

Q → (a)

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/enm.h>
#include <sys/mman.h>
#include <sys/stat.h>
#include <time.h>
#include <unistd.h>
#include <sys/wait.h>
```

```
#define BUFFER 128
```

```
char strA[BUFFER+1];
char strB[BUFFER+1];
```

```
void gen_message(char *str)
```

```
{
    srand(time(NULL) ^ getpid());
    for (int i = 0; i < BUFFER-4; i++)
    {
        str[i+4] = 'A' + rand() % 26; // gen message
    }
    strA[BUFFER] = '\0';
    sprintf(str, "msg: %s", str+4);
}
```



```

void gen_message B(char * str)
{
    srand(time(NULL) ^ getpid());
    for(int i=0; i<Butler-4; i++)
    {
        str[i+4] = 'A' + rand() % 26;
    }
    str[Butler] = '\0';
    printf(str, "message B %s", str+4);
}

int main()

```

```

{
    const int size = 4096;
    const char * name = "prod-cons";
    int fd;
    char * ptr;
    pid_t consumers[2];
    for(int i=0; i<2; i++)
    {
        consumers[i] = fork();
        if(consumers[i] < 0)
        {
            perror("Error finding consumers");
            exit(1);
        }
        else if (consumers[i] > 0)
        {
            fd = shm_open(name, O_CREAT | O_RDWR,
                truncate(fd, size);
            ptr = (char *) mmap(0, size, PROT_WRITE, MAP_SHARED,
                fd, 0); // from Internet.

```

```

    gen_message A(str A);
    gen_message B(str B);

```



```

    sprintf(ptr, "%s", strA);
    ptr += strlen(strA);
    sprintf(ptr, "%s", strB);
    // Producer exits loop.
    break;
else if (consumer[i] == 0)
{
    fd = shm_open(name, O_RDONLY, 0666);
    ptr = (char *) mmap(0, SIZE, PROT_READ, MAP_SHARED, fd, 0);

    if (i == 0)
    {
        if (strcmp(ptr, "Money A") == 0)
        {
            printf("Con A: %s", ptr);
        }
    }

    else if (i == 1)
    {
        if (strcmp(ptr + 4, "Money B", 4) == 0)
        {
            printf("Consumer B: %s", ptr + 4);
        }
    }

    break;
}

for (i = 0; i < 2; i++)
{
    wait(NULL);
}
}

```



~~shm unlink~~

Date: \_\_\_\_\_

```
shm unlink(name);  
return 0;
```

```
}
```

(b)

```
#include <stdio.h>
```

```
/* Input Lib. */
```

```
#include <sys/mman.h>
```

```
#define BUFFER 120
```

```
void Consumer_mrges (const char *name)
```

```
{
```

```
    const int size = 4096;
```

```
    int fd; char *ptr
```

```
    fd = shm_open(name, O_RDONLY, 0666);
```

```
    if (fd == -1)
```

```
    {  
        perror("Error opening shared memory");  
        exit(1);  
    }
```

```
    ptr = (char *) mmap(0, size, PROT_READ,  
        MAP_SHARED, fd, 0);
```

```
    if (ptr == MAP_FAILED)
```

```
    {  
        perror("Error in mapping");  
        exit(1);  
    }
```

```
}
```



```

while(1)
{
    if(strlen(ptr) > 0)
    {
        printf("Received message: %s", ptr);
        memset(ptr, 0, size);
    }
    usleep(10000);
}

```

```

if(mmap(ptr, size) == -1)
{
    perror("Error mapping shared mem");
    exit(1);
}

```

```

if(close(fd) == -1)
{
    perror("Error closing shared mem");
    exit(2);
}
}

```

```

int main()
{
    const char *name = "/proc/cpuinfo";
    pid_t consumer_pid = fork();
}

```

```

if(consumer_pid == -1)
{
    perror("Error forking consumer");
    exit(2);
}

```

```

else if(consumer_pid == 0)
{
    consumer_pid = getpid();
    exit(0);
}

```

```

wait(NULL);
return 0;
}

```