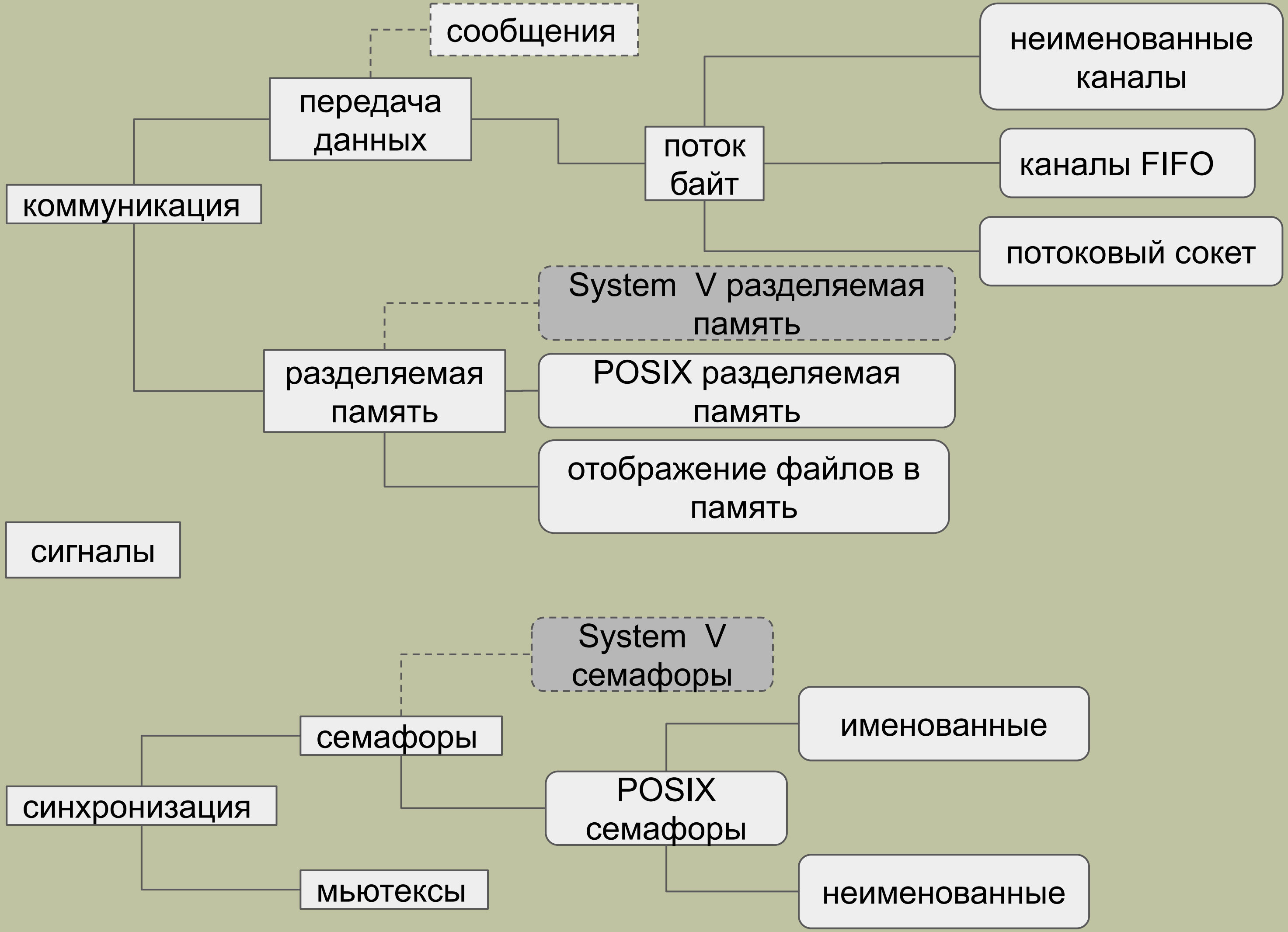
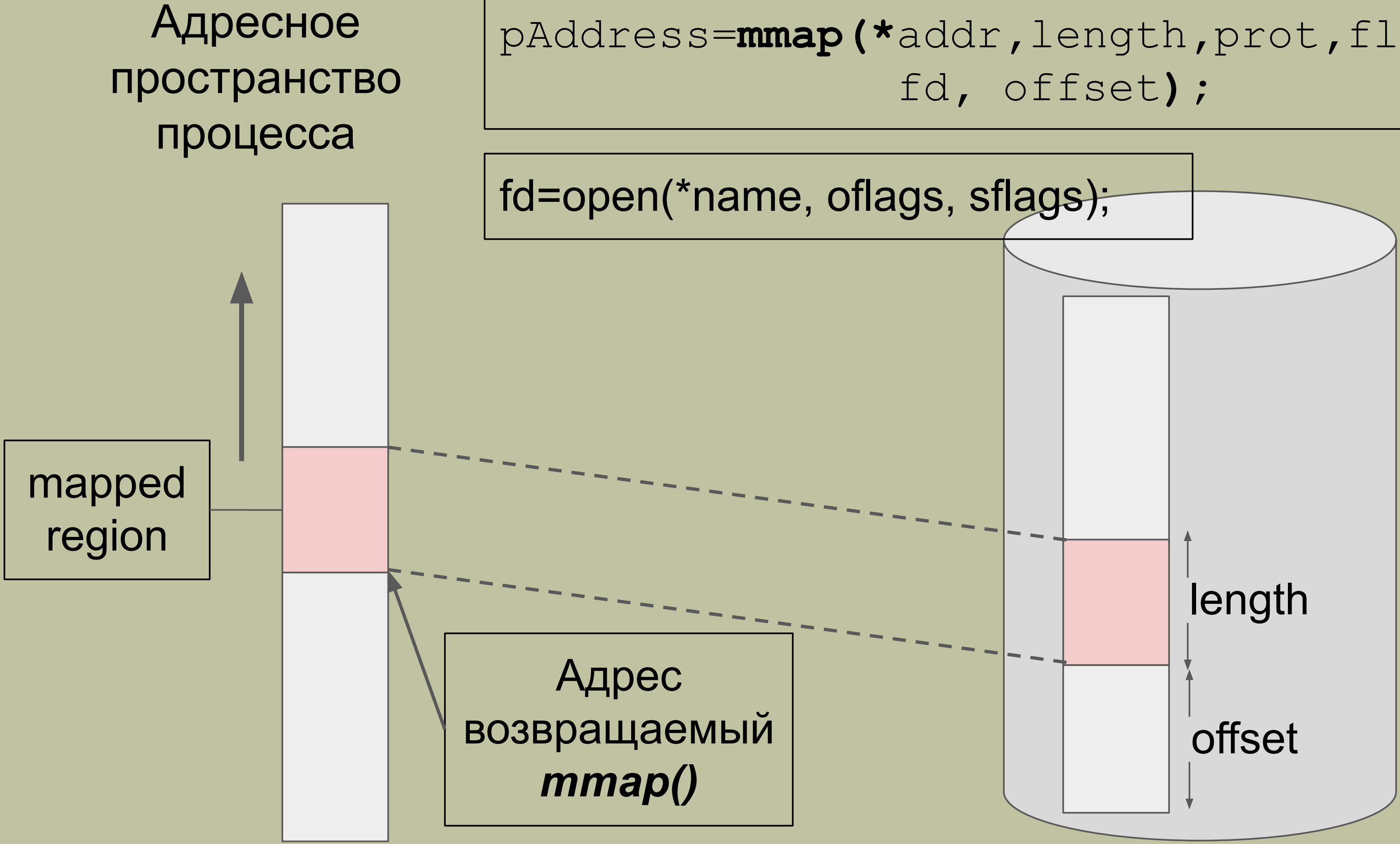
Лекция 12

- Межпроцессное взаимодействие (*IPC*).
- Отображение файлов в память.
- Выделение разделяемой памяти. Именованные семафоры *POSIX*.


сообщения
передача данных
поток байт
неименованные каналы
каналы FIFO
потоковый сокет
коммуникация
System V разделяемая память
разделяемая память
POSIX разделяемая память
отображение файлов в память
сигналы
System V семафоры
семафоры
именованные
POSIX семафоры
синхронизация
неименованные
мьютексы

```
pAddress=mmap(*addr,length,prot,flags,
                    fd, offset);
```

```
fd=open(*name, oflags, sflags);
```



prot: PROT_READ, PROT_WRITE, PROT_EXEC

flags: MAP_PRIVATE, MAP_SHARED

oflags: O_RDONLY, O_WRONLY, O_RDWR, O_CREAT, O_APPEND,...

sflags: S_IRUSR, S_IWUSR, S_IXUSR,
S_IRGRP, S_IWGRP, S_IXUSR,
S_IROTH, S_IWOTH, S_IXOTH

```
> ls -l
-rw-r--r-- 1 malkov users    69 Dec  6 12:16 s_test
```

```
>chmod  760 s_test
>ls -l
-rwxrw---- 1 malkov users    69 Dec  6 12:17 s_test
```

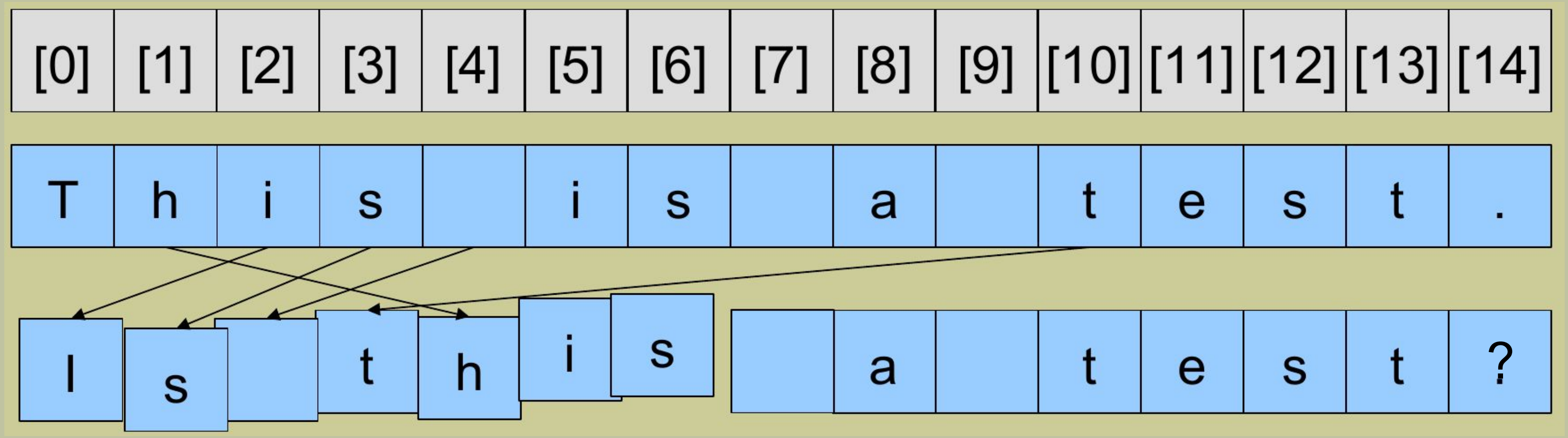
S_IRUSR | S_IWUSR | S_IXUSR |
S_IRGRP, S_IWGRP

```
int fd;          //дескриптор файла
FILE* fp;    //файловый поток (указатель на структуру)
fd=open(*name, oflags, sflags);
read(fd,...);
write(fd,…);
lseek(fd,...);
fp=fopen(*name, mode); //mode: r,w,a,r+w+
fread(...,fp);
fwrite(…,fp);
fseek(fp,...);
fd=fileno(fp);
fp=fdopen(fd, mode);
close(fd);
fclose(fp);
```

lab13b.c

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

int main(int argc, char* const argv[]){
  int fd;
  struct stat stat_file;
  char dummy;
  char* map_address;
```

```
fd=open("test.txt", O_RDWR | O_CREAT,
          S_IRUSR | S_IWUSR | S_IRGRP);
  if (fd == -1)
    fprintf(stderr, "open\n");

  if(fstat(fd, &stat_file))
    fprintf(stderr, "fstat\n");

  map_address=(char*)mmap(0,stat_file.st_size,
                   PROT_READ | PROT_WRITE,
                   MAP_PRIVATE, fd, 0);

  if (map_address == MAP_FAILED)
    fprintf(stderr, "mmap\n");
```



```
dummy=map_address[1];
map_address[0]=map_address[5]-0x20;
map_address[1]=map_address[3];
map_address[2]=map_address[4];
map_address[3]=map_address[10];
map_address[4]=dummy;
map_address[14]=63;
```

```
  write(fd, map_address, stat_file.st_size);

  munmap(map_address, stat_file.st_size);

  close(fd);

  return 0;
}
```

lab13c-1.c

```
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>
#include <sys/mman.h>

int main(int argc, char* const argv[]){
  int fd;
  struct stat stat_file;
  char* map_address;

  fd=open("test_shared.txt", O_RDWR | O_CREAT,
                              S_IRUSR | S_IWUSR | S_IRGRP);
  if (fd == -1)
    fprintf(stderr, "open\n");
```

```
 map_address=(char*)mmap(0,256,
                PROT_READ | PROT_WRITE,
                MAP_SHARED, fd, 0);
 if (map_address == MAP_FAILED)
   fprintf(stderr, "mmap\n");
 close(fd);

 memcpy(map_address, "Take it easy!\0", sizeof("Take it easy!\0"));

 getc(stdin);

 munmap(map_address, 256);

 return 0;
}
```

lab13c-2.c

```
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>
#include <sys/mman.h>

int main(int argc, char* const argv[]){
  int fd;
  struct stat stat_file;
  char* map_address;

  fd=open("test_shared.txt", O_RDWR);
  if (fd == -1)
    fprintf(stderr, "open\n");
```

```
 map_address=(char*)mmap(0,256,
              PROT_READ | PROT_WRITE,
              MAP_SHARED, fd, 0);
 if (map_address == MAP_FAILED)
   fprintf(stderr, "mmap\n");
 close(fd);

 write(fileno(stdout), map_address, 256);

 getc(stdin);

 munmap(map_address, 256);

 return 0;
}
```

```
~/Лекция12> cat /proc/7704/maps
00400000-00401000 r-xp 00000000 08:13 1690022666                    ~/lab13c-1
00600000-00601000 r--p 00000000 08:13 1690022666                    ~/lab13c-1
00601000-00602000 rw-p 00001000 08:13 1690022666                    ~/lab13c-1
01e30000-01e51000 rw-p 00000000 00:00 0                             [heap]
………………………… …………………………………………………………………..
7f2aad407000-7f2aad42c000 r-xp 00000000 00:2d 671128                /lib64/ld-2.26.so
7f2aad62b000-7f2aad62c000 rw-s 00000000 08:13 1690022692            ~/test_shared.txt
7f2aad62c000-7f2aad62d000 r--p 00025000 00:2d 671128                /lib64/ld-2.26.so
……………………………………………………………………………………………….
```

```
~/Лекция12> cat /proc/7969/maps
00400000-00401000 r-xp 00000000 08:13 1690022681                    ~/lab13c-2
00600000-00601000 r--p 00000000 08:13 1690022681                    ~/lab13c-2
00601000-00602000 rw-p 00001000 08:13 1690022681                    ~/lab13c-2
01a1c000-01a3d000 rw-p 00000000 00:00 0                             [heap]

7f769bd6e000-7f769bd93000 r-xp 00000000 00:2d 671128                /lib64/ld-2.26.so
7f769bf5c000-7f769bf5e000 rw-p 00000000 00:00 0
7f769bf92000-7f769bf93000 rw-s 00000000 08:13 1690022692            ~/test_shared.txt
7f769bf93000-7f769bf94000 r--p 00025000 00:2d 671128                /lib64/ld-2.26.so
```

lab13d-1.c

```
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <unistd.h>
 #include <string.h>
#include <sys/mman.h>

int main(int argc, char* const argv[]){
  int fd;
  char* map_address;

  fd=shm_open("/common_region",
              O_RDWR | O_CREAT,
              S_IRUSR | S_IWUSR | S_IRGRP);
  if (fd == -1)
    fprintf(stderr, "open\n");
  ftruncate(fd, 256);
  map_address=(char*)mmap(0,256,
                  PROT_READ | PROT_WRITE,
                  MAP_SHARED, fd, 0);
  if (map_address == MAP_FAILED)
    fprintf(stderr, "mmap\n");
  close(fd); q

  memcpy(map_address, "Take it easy! Be happy!\0",
             sizeof("Take it easy! Be happy!\0"));

  getc(stdin);

  munmap(map_address, 256);
  shm_unlink("/common_region");

  return 0;
}
```

```
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>
#include <sys/mman.h>

int main(int argc, char* const argv[]){
  int fd;
  char* map_address;

  fd=shm_open("/common_region", O_RDWR,
             S_IRUSR | S_IWUSR | S_IRGRP);
  if (fd == -1)
    fprintf(stderr, "shm_open\n");
```

lab13d-2.c

```
  map_address=(char*)mmap(0,256,
                  PROT_READ | PROT_WRITE,
                  MAP_SHARED, fd, 0);
  if (map_address == MAP_FAILED)
      fprintf(stderr, "mmap\n");
  close(fd);

  write(fileno(stdout), map_address, 256);

  getc(stdin);

  munmap(map_address, 256);

  return 0;
}
```

```
mapshm> gcc lab13d-1.c -lrt -o lab13d-1
```

```
~/Лекция12> cat /proc/7896/maps
00400000-00401000 r-xp 00000000 08:13 32664504              ~/lab13d-1
00600000-00601000 r--p 00000000 08:13 32664504              ~/lab13d-1
00601000-00602000 rw-p 00001000 08:13 32664504              ~/lab13d-1
01d94000-01db5000 rw-p 00000000 00:00 0                      [heap]
7f5e8071d000-7f5e80736000 r-xp 00000000 00:2d 671162
/lib64/libpthread-2.26.so

7f5e80efe000-7f5e80eff000 rw-p 00007000 00:2d 671166          /lib64/librt-2.26.so
7f5e80eff000-7f5e80f24000 r-xp 00000000 00:2d 671128          /lib64/ld-2.26.so
7f5e810ea000-7f5e810ef000 rw-p 00000000 00:00 0
7f5e81123000-7f5e81124000 rw-s 00000000 00:17 132423          /dev/shm/common_region
7f5e81124000-7f5e81125000 r--p 00025000 00:2d 671128          /lib64/ld-2.26.so
7f5e81125000-7f5e81126000 rw-p 00026000 00:2d 671128          /lib64/ld-2.26.so
```

lab13em-1.c

```
#include <pthread.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/mman.h>

int main( void ) {
  int n=0;
  int fd;
  char* sh;
  pthread_mutex_t* Mutex;
  pthread_mutexattr_t mutex_attr;
```

```
fd=shm_open("/common_region1",
            O_RDWR | O_CREAT,
            S_IRUSR | S_IWUSR | S_IRGRP);
 if (fd == -1)
   fprintf(stderr, "shm_open\n");
 ftruncate(fd, 6);



 sh=(char*)mmap(0,6,
                PROT_READ | PROT_WRITE,
                MAP_SHARED, fd, 0);
 if (sh == MAP_FAILED)
    fprintf(stderr, "mmap\n");
 close(fd);
 memset(sh,0,6);
```

```
fd=shm_open("/common_mutex",
             O_RDWR | O_CREAT,
             S_IRUSR | S_IWUSR | S_IRGRP);
  if (fd == -1)
    fprintf(stderr, "shm_open for mutex\n");
  ftruncate(fd, sizeof(pthread_mutex_t));

  pthread_mutexattr_init(&mutex_attr);
  pthread_mutexattr_setpshared(&mutex_attr, PTHREAD_PROCESS_SHARED);
  Mutex=(pthread_mutex_t*)mmap(0,sizeof(pthread_mutex_t),
                           PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
  close(fd);

  pthread_mutex_init(Mutex, &mutex_attr);
```

```
 while(1){
       pthread_mutex_lock(Mutex);
              //write(fileno(stdout),sh, 6);
              printf("String: %s\n",sh);
       pthread_mutex_unlock(Mutex);
  }

  munmap(sh, 6);
  munmap(Mutex, sizeof(pthread_mutex_t));
  shm_unlink("/common_mutex");
  shm_unlink("/common_region1");

  getc(stdin);
  return 0;
}
```

lab13em-2.c

```
#include <pthread.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/mman.h>

int main(){
  int n=0;
  int counter=0;
  int fd;
  char* sh;
  pthread_mutex_t*  Mutex;
```

```
fd=shm_open("/common_region1",
            O_RDWR | O_CREAT,S_IRUSR | S_IWUSR | S_IRGRP);
 sh=(char*)mmap(0,6,
                 PROT_READ | PROT_WRITE,
                 MAP_SHARED, fd, 0);
 close(fd);
 memset(sh,0,6);

 fd=shm_open("/common_mutex",
            O_RDWR | O_CREAT,
            S_IRUSR | S_IWUSR | S_IRGRP);



 Mutex=(pthread_mutex_t*)mmap(0,sizeof(pthread_mutex_t),
                             PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
 close(fd);
```

```
while ( 1 ){
 pthread_mutex_lock(Mutex);
      if(counter%2){
             sh[0]='H';sh[1]='e';sh[2]='l';sh[3]='l';sh[4]='o';sh[5]='\0';
      }
      else{
             sh[0]='B';sh[1]='y';sh[2]='e';sh[3]='_';sh[4]='u';sh[5]='\0';
      }
 pthread_mutex_unlock(Mutex);
 counter++;
}
getc(stdin);
```

```
 munmap(sh, 6);
 shm_unlink("/common_region1");
 munmap(Mutex, sizeof(pthread_mutex_t));
 shm_unlink("/common_mutex");



 return 0;
}
```

lab13e-1.c

```
#include <semaphore.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/mman.h>

int main( void ) {
  int n=0;
  int fd;
  char* sh;
  sem_t  *sem;
```

```
fd=shm_open("/common_region",
             O_RDWR | O_CREAT,
             S_IRUSR | S_IWUSR | S_IRGRP);
 if (fd == -1)
   fprintf(stderr, "shm_open\n");

 ftruncate(fd, 6);

 sh=(char*)mmap(0,6,
                 PROT_READ | PROT_WRITE,
                 MAP_SHARED, fd, 0);
 if (sh == MAP_FAILED)
    fprintf(stderr, "mmap\n");

 memset(sh,0,6);
```

```
sem=sem_open("/common_sem", O_CREAT,
             S_IRUSR | S_IWUSR | S_IRGRP, 1);
if (sem == SEM_FAILED)
  fprintf(stderr, "sem_open");

while(n++<200){
 sem_wait(sem);
  //write(fileno(stdout),sh, 6);
 printf("String: %s\n",sh);
 sem_post(sem);
 usleep(100);
}
```

```
  shm_unlink("/common_region");
  munmap(sh, 6);

  sem_unlink("/common_sem");
  sem_close(sem);
  return 0;
}
```

```
mapshm> gcc lab13e-1.c -lpthread -lrt -o lab13e-1
```

```
mapshm> gcc lab13e-2.c -lpthread -lrt -o lab13e-2
```

```
#include <semaphore.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/mman.h>

int main(){
  int n=0;
  int counter=0;
  int fd;
  char* sh;
  sem_t  *sem;
```

lab13e-2.c

```
fd=shm_open("/common_region",
            O_RDWR | O_CREAT,S_IRUSR | S_IWUSR | S_IRGRP);

 sh=(char*)mmap(0,6,
                PROT_READ | PROT_WRITE,
                MAP_SHARED, fd, 0);
 memset(sh,0,6);



 sem=sem_open("/common_sem", 0);
```

```
while ( n++<200 ){
   sem_wait(sem);
   if(counter%2){
    sh[0]='H';sh[1]='e';sh[2]='l';sh[3]='l';sh[4]='o';sh[5]='\0';
  }
  else{
    sh[0]='B';sh[1]='y';sh[2]='e';sh[3]='_';sh[4]='u';sh[5]='\0';
  }
  sem_post(sem);
  counter++;
  usleep(100);
 }
 munmap(sh, 6);
 return 0;
}
```

```
00400000-00401000 r-xp 00000000 08:13 21720846   ~/lab13e-2
……………………………………………………………………………
………………………………………………
7f34c6dee000-7f34c6def000 rw-s 00000000 00:17 79255
                                                      /dev/shm/sem.common_sem
7f34c6def000-7f34c6df0000 rw-s 00000000 00:17 79254
                                                      /dev/shm/common_region
......................................................................................................................................................
```

```
/Лекция12> ls -ltr /dev/shm
-rw------- 1 wwwrun www   54440 Nov 12 11:58 ShM.9199a53eH348827e0
-rw------- 1 wwwrun www    4096 Nov 12 11:58 mono.1868
-rw-r----- 1 malkov users    32 Nov 12 15:18 sem.common_semap
-rw-r----- 1 malkov users     6 Nov 12 15:18 common_region
```