1	2	3	4

Контрольная работа №3.

Вариант 1. *(ФИО: Казырид М. И.)*

Задание 1. Написать программу.

Организовать у процесса-«отца» двух «сыновей». Старший сын должен уметь выводить на экран предложение *«Воображение важнее,»*. Младший - *« чем знание. »*. В совместной работе сыновья должны вывести в stdout 10 раз сообщение: *«Воображение важнее, чем знание. »*, - и умереть. Отец должен «умереть» после «смерти» сыновей.

Замечания: Механизм организации «смерти» произвольный. Синхронизацию работы «сыновей» реализовать с помощью ***каналов****.*

Задание 2. Написать программу.

После запуска программы процесс должен начать бесконечно выводить строку: *«Значимость программы »*. В случае нажатия пользователем на клавиатуре «Ctrl+C» программа должна вывести строку *«обратно пропорциональна »*. И начать бесконечно выводить строку *«объему выводимых ею результатов »*. В случае повторного нажатия на «Ctrl+C» процесс должен умереть от полученнного **сигнала**.

Задание 3. Написать две программы.

Программа-чтец должна организовать **IPC очередь сообщений**, передать в неё содержимое текстового файла «PeaceAndWar.txt» построчно и завершить своё исполнение.

Программа-писарь должна прочитать сообщения из общей с «чтецом» очереди сообщений, вывести их на экран и закончить своё исполнение.

Примечание: работу с файлом реализовать любым удобным Вам образом.

Задание 4. Написать две программы.

Первая программа должна выводить на экран заглавные буквы английского алфавита от ‘Z’ до ‘A’ в обратном алфавитном порядке.. Вторая программа – прописные от ‘z’ до ‘a’ (в обратном алфавитном порядке). С помощью **семафоров IPC** организовать синхронизацию выполнения программ так, чтобы на экранее заглавные и прописные буквы выводились строго по очереди.

Например, правильно: Zz Yy Xx ... Bb Aa

Неправильно: ZY zy xX ... Bb aA

ФИО: Казырид Максим

№2

```
int old=1, new=0;
void Fpr(int sigNum)
{ signal(SIGINT, SIG_DFL);
  new++;
}
int main()
{
  signal(SIGINT, Fpr);
  while (old!=new)
  { printf("Значимость программы");
  }
  printf("обратно пропорциональна");
  while(1){
    printf("объему выводимых его результатов");
  }
  return 0;
}
```

№1

```
int main()
{ int fd1[2], fd2[2], i=0, k, j;

  pipe(fd1);
  pipe(fd2);
  if (fork()==0)
  {
    close(fd1[0]); close(fd2[1]);
    while (i<10)
    { printf("Воображение важнее");
      i++;
      write(fd1[1], &i, sizeof(int));
      read (fd2[0], &k, sizeof(int));
    }
    close(fd1[1]);
    close(fd2[0]);
    exit(0);
  } else
```

```
if (fork()==0)
{  close(fd2[0]);
   close(fd1[1]);
   while (i<10)
   {  read(fd1[0],&j,sizeof(int));
      printf("ребенок значение.\n");
      write(fd2[1],&i,sizeof(int));
   }
   close(fd2[1]);
   close(fd1[0]);
   exit(0);
} else    //отец
{  close(fd1[0]);
   close(fd1[1]);
   close(fd2[0]);
   close(fd2[1]);
   wait();
   wait();
}
return 0;
}
```

+

Не выделена память,

segmentation fault

№3

отец

```
int main()
{ int id1, id2
  char *string;
  struct Buf *PS;
  PS=(...);
*PS.string=(char*)malloc(256*sizeof(char));
  id1=ftok("/user/bin/a.txt",'a');
  msgget(id1,0,0);
  id2=open("PeaceAndWar.txt",O_RDONLY,0777);
  while (fgets(PS.string,256,id2)>0)
  {  msgsnd(id1,PS.string,0);
  }
  close(id2);
  msgctl(id1,IPC_RMID);
  return 0;
}
struct Buf{
     long k;
     char *string;
}
```

сколько?

сын

```
int main()
{ int id1;
  struct Buf *PS;
*PS.string=(char*)malloc(256*sizeof(char));
  id1=ftok("/user/bin/a.txt",'a');
  while (msgrcv(id1,PS,0)>0)
     printf("%s",*PS.string);
  return 0;
}
```

msgget ??

—

Не подключились к IPC

—

№4

```
int main()
{ int id1,s=1;
  char c='Z';
  id1=ftok("/use1/bin/a.txt",'a');
  semget(id1,1,0);
  while (c!='A')
  {
     semop(1,s,1);
     printf("%c",c);
     c=c-1;
     semop(1,s,-1);
  }
```

```
int main()
{ int id1;
  char c='z';
  id1=ftok("/user/bin/a.txt",'a');
  while (c!='a'-1)
  {
     semop(1,s,1);
     printf("%c ",c);
     c=c-1;
     semop(1,s,-1);
  }
  return 0;
```

semget?

Казырид Максим

ФИО: Черновик

№2

```
int main()
{ int old=1, new=0;
  signal(SIGINT, FP2);
  while (old!=new)
  {
    printf("значимость программы ");
  }
  printf("обратно пропорциональна");
  signal(SIGINT, SIG_DFL);
  while(1){
    printf("обычному выводимому результату");
  }
  return 0;
}

void fp2()
{
  new++;
}
```

№1

```
int main()
{ int fd1[2], fd2[2];
  int k=0, id1, id2;
  char c;
  pipe(fd1);   c='a';
  pipe(fd2);
  if ((id1=fork())==0)
  { close(fd1[0]); close(fd2[1]);
    printf("Воображение важнее");
    while(read(fd1[1], c, 1));
    write(fd2[0], c, 1);
  } close(fd1[1]); close(fd2[0]);
  exit(0);
}
else
if ((id2=fork())==0)
{ close(fd2[0]); close(fd1[1]);
```

while (i<10)

while (read(fd2[0], i)==

```
else
{ wait();
  wait();
  return 0;
}
}
```

№5

```
int main( )
{  int id1, a2
   char *string;
   string=(char*)malloc(256*sizeof(char));
   id1=ftok("a.txt", 0644|O_CREAT);
   msgget(id1
   fd=open("peace
while(fgets(string, 256,
   msgsnd(id1, string
   close(id1);
   return 0;
}
int main( )
{  int id1;
   char *string;
   string=(char*)malloc(256*sizeof(char));
   id1=ftok("a.txt", 0644|O_CREAT);
   msgget(id1, );
   while(msgrcv(id1, string, );
      printf("%s", string));
   return 0
}
```

№4

```
int main()
{ int id1
  char c='z'
  id1=ftok("a.txt",
  while(c!='A')
  { semop( );
    printf("%c", c))
    c=c-1)
  }
return 0;
```

```
int main()
{ int id1;
  char c='z';
  id1=ftok("a.txt", 'a');
  semget(id1, 0
  while(c!='a')
  { semop(
    printf("%c", c);
    c=c-1;
  }
return 0;
```