

1	2	3	4
0	+	+	+

Контрольная работа №3.

Вариант 2. (ФИО: Курмов А.А. 21121)

Задание 1. Написать программу.

У процесса-«деда» (далее «Д») создать «внука» (далее «В»). «Д» должен передать «В» два вопросительных предложения, придуманных Вами. После получения каждого из предложений «В» должен вывести их в stdout, прочитать из stdin строку-«ответ» и отправить эту строку к «Д».

После передачи последнего «ответа» «В» должен умереть. «Д» после получения последнего ответа должен нормально закончить свою работу (умереть сам).

Замечания: Механизм организации «смерти» «В» произвольный. Синхронизацию работы «Д» и «В» реализовать с помощью каналов. «Д» посылает второй «вопрос» только после получения «ответа» от «В» на первый.

Задание 2. Написать программу.

Напишите программу, которая будет ждать получения **сигналов**. Причём на один из сигналов (например, SIGINT) программа должна вывести в stderr: «Отстань». Получение другого типа сигналов программа должна игнорировать. А на третий тип сигналов она должна написать в stdout: «Согласна», - и закончить своё выполнение.

Замечания: Можно использовать функции ввода/вывода как высокого, так и низкого уровней. Механизм окончания выполнения программы – произвольный. Тип сигналов первого, второго и третьего вида – на Ваш выбор.

Задание 3. Написать две программы.

Программа-сервер должна запросить у пользователя целое положительное число N, организовать **IPC очередь сообщений**, передать в неё N любых нечётных знаковых целых чисел и завершить своё исполнение.

Программа-клиент должна прочитать все N чисел из общей очереди сообщений, записать их в файл «statistika.txt», уничтожить IPC очередь сообщений и закончить своё исполнение.

Задание 4. Написать две программы.

Первая программа должна выводить на экран нечётные числа от 1 до 99 (в порядке возрастания). Вторая программа – чётные от 2 до 98 (в порядке возрастания). С помощью **семафоров IPC** организовать синхронизацию выполнения программ так, чтобы на экране все числа от 1 до 99 выводились строго в порядке их возрастания.

Например, правильно: 1 2 3 4 ... 96 97 98 99

Неправильно: 1 3 2 4 ... 96 97 99 98

ФИО: Кермов

(N4)

```

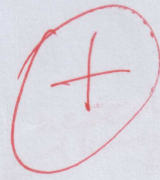
int main()
{

```

```

int main()
{
    int id_ipc, id_sem;
    int i;
    id_ipc = ftok("first", 'f');
    id_sem = semget(id_ipc, 2, 0666 | IPC_CREAT);
    struct sembuf hang;
    struct sembuf next;
    next.sem_flg = 0;
    hang.sem_flg = 0;
    next.sem_op = 1;
    next.sem_num = 1;
    hang.sem_op = -1;

```



```

    hang.sem_num = 1;
    next.sem_num = 2;

```

(1)

```

    hang.sem_num = 2;
    next.sem_num = 1;
    semop(id_sem, &hang, 1); // because

```

(2)

```

for (i = 1; i <= 97; i += 2)
{
printf("%d\n", i);

```

copy past ← →

```

for (i = 2; i <= 96; i += 2)
{
    printf("%d\n", i);
    semop(id_sem, &next, 1); // because
    semop(id_sem, &hang, 1); // because

```

```

}
printf("%d\n", i); // because
semctl(id_sem, 0, IPC_RMID, (int)0);
return 0;
}

```

```

}
printf("%d\n", i); // 98.
semop(id_sem, &next, 1);
return 0;
}

```

}

}

SIGINT
SIGKILL
SIGUSR1
SIGSTOP²

ФНО: Корнев

2

√2

write(~~fd~~, &buf, size);

```

int
void int_action(int i)
{
printf("omama\n");
signal(SIGINT, int_action);
}
void kill_action(int i)
{
printf("omama\n");
exit(0);
}
int main()
{
signal(SIGINT, int_action);
signal(SIGKILL, kill_action);
signal(SIGUSR1,
while (1) { }
return 0;
}

```

```

char *msg = "omama\n";
write(2, msg, 8);

```



SIG-IGN

Меню непереносит !!!
 (through)

√3

Циклы:

Классика - см. предыдущий лист

```

int main()
{
int n;
int id_ipc, id_msg;
int i;
i = 1;
int j;
scanf("%d", &n);
id_ipc = ftok("server", 's');
id_msg = msgget(id_ipc, 0666|IPC_CREAT);
for (j=0; j<n; j++)
{
sms.value = i;
msgsnd(id_msg, (struct msgbuf *)(&sms), sizeof(int), 0);
i += 2;
i = -i;
}
return 0;
}

```

```

struct int_msg
{
long type;
int value;
};

```

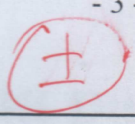
```

struct int_msg sms;
sms.type = 1;
sms.value = n;
msgsnd(id_msg, ...);

```

// комментарий к
 см. ниже.

ФИО: Корнилов



№3

Клиент:

struct int_msg
отличается в сервере

```
// объявление файла
FILE *file;
file = fopen("statistika.txt", "w");
```

int main()

```
{
  int id_ipc, id_msg;
  struct int_msg sms;
  int i;
  int n;
  int trash;
  id_ipc = atoi("server", 's');
  id_msg = msgget(id_ipc, 0666);
  if (id_msg == -1)
    exit(0);
```

```
msgrcv(id_msg, (struct msgbuf *)(& sms), sizeof(int), 1, 0); (1)
  n = sms.value;
  for (i = 0; i < n; i++)
  {
    (1) sms ←
    test = sms.value;
```

```
write(file, &test, sizeof(int));
```

write ← нужен уровень, не работает с FILE

```
fclose(file);
msgctl(id_msg, IPC_RMID, NULL);
return 0;
```

Удалить данные
ТОТ, что сейчас ресурс
(хороший топ)