

Задачи за упражнение за контролно 2

Тестови

1. С кой системен примитив от IPC пакета на UNIX System V можем да маркираме за унищожаване shared memory segment?
2. С кой системен примитив от IPC пакета на UNIX System V можем да създадем message queue?
3. С кой системен примитив от IPC пакета на UNIX System V можем да променим съдържанието на shared memory segment?
4. С кой системен примитив от IPC пакета на UNIX System V можем да променим стойностите в масив от семафори?
5. Как трябва да извикаме системния примитив shmget, за да създадем нов shared memory segment с външно име, записано в променливата key_t key, който да побира променлива от тип struct mystr, където дефиницията на структурата е:

```
struct mystr
{
    long a;
    char b[20];
    int n[10];
};
```

6. Как трябва да извикаме системния примитив shmat, за да добавим към адресното пространство на процеса shared memory segment с вътрешен идентификатор, записан в променливата int shmid, така, че достъпът на нашия процес до този сегмент да е write-only?
7. Какво се подава като втори аргумент на системния примитив shmget?
8. Какво се подава като втори аргумент на системния примитив shmctl?
9. Какво ни връща при успех системния примитив msgget?

С pipe:

1. Да се напише програма, която да приема два аргумента – валидно име на файл filename и положително число N , и извежда на

стандартния изход номерирани първите N реда от файла.

Използвайте `cat`, `head` и др.

2. Да се напише програма, която чете редове от стандартния вход, които имат формат

```
cmd1 arg1 arg2 | cmd2 brg1 brg2
```

(всички думи в реда са разделени точно с по един интервал)

и пуска за изпълнение съответния `pipe` от команди.

Със `shared memory`:

1. Да се напише програма, която създава сегмент обща памет, побиращ `int`, и създава процес-дете. По 10000 пъти процесът-дете и процесът-родител променят стойността на този сегмент, започвайки от стойност 0, като родителят я увеличава с 1, а детето – с 5. Всеки процес изписва на стандартния изход своя пид и новата стойност след всяка промяна.

Забележка: Не се грижим за евентуален конкурентен достъп. Но се грижим за унищожаването на сегмента!

2. Да се напише програма, която създава сегмент обща памет, побиращ 20 символа, зарежда в тях стойност 'A' и създава процес-дете. По 10 пъти процесът-родител и процесът-дете променят стойността на произволен от тези символи, като родителят я увеличава с 1, а детето – с 2. Всеки процес да изписва на стандартния изход своя пид и новата стойност след всяка промяна.

Забележка: Не се грижим за евентуален конкурентен достъп. Но се грижим за унищожаването на сегмента!