

### Задания к лабораторной работе №1 по курсу “Системное программирование”

1. Написать на си++ программу для удаления комментариев и тегов из html-файла. Обеспечить правильную подстановку для знаков <, >, & и ". Использовать регулярные выражения библиотеки `pcresrr` и объектный ввод-вывод.
2. Реализовать на си++ классы “потребитель” и “производитель”, реализующие соответствующие процессы. Производитель должен генерировать случайные числа в максимально допустимом функцией `rand` диапазоне. Потребитель должен проверять числа на простоту, вести статистическую обработку (всего чисел обработано и доля простых) и печатать ее текущие результаты. Производитель должен после генерации 10000 чисел делать паузу на 1 секунду. Программа должна прекращаться по перехватываемому сигналу `SIGINT`. При реализации использовать стандартный контейнер очередь, сопроцессы-нити и средства управления ими, мьютексы. Сравнить результат с теоретическим, т.е. с теоретической долей простых чисел в заданном диапазоне.
3. Сделать распределитель памяти, который более эффективно, чем стандартный, выделяет память для скалярных данных. Такой распределитель должен выделять память кусками, с запасом для нескольких скалярных единиц. Распределитель должен быть совместим с контейнерами стандартной библиотеки си++. В распределителе должна быть функция `showMemoryMap`, показывающая текущее распределение памяти, например, в следующем далее виде.

```
Chunk #0
    10 link(s) occupied
    1001 link(s) free
    22 link(s) occupied
Chunk #1
    500 link(s) free
    16 link(s) occupied
...
```

Рекомендуется использовать при реализации этой функции массивы бит. Распределение памяти для массивов (и даже скаляров с кратными 16 байт размерами) можно переадресовывать стандартным средствам. Готовая программа должна содержать контейнер `map` с заданным распределителем.

### Задания к лабораторной работе №2 по курсу “Системное программирование”

1. Реализовать на ассемблере расчёт факториала для аргумента до 34 включительно, т. е для результата до 128 бит. Обеспечить интерактивный ввод аргумента.
2. Обеспечить ввод 3-х строк символов. Найти все вхождения 2-й строки в 1-ю и заменить их на 3-ю строку. Вывести строку-результат и количество проведенных замен на экран дисплея. Использовать строковые инструкции ассемблера. Не использовать сторонние библиотеки.
3. Сделать предыдущее задание на ассемблере через обращения к средствам стандартной библиотеки языка си для ввода-вывода и операций со строками.
4. Написать на си++ программу, печатающую значение каждого значащего бита регистра `SR`. Должно печататься имя бита и его значение. Должно быть также напечатано 16-чное значение `SR`. Можно использовать подпрограмму на ассемблере или средства оператора `asm`.
5. Написать макрос `prfmt`, который бы соответствовал функции `printf` стандартной си-библиотеки. Пример вызова,

```
prfmt msg,8,10,12,14,16,18,20,txt
```

в качестве `msg` и `txt` могут быть, например,

```
msg db "Current numbers are %d, %d, %d, %d, %d, %d, %d - %s.",0xA,0
txt db "finish",0
```

Использовать вызов упомянутой функции `printf`. Поддержка строковых констант в списке аргументов не обязательна.