

**Вступительные задания на летнюю практику в АО «МЦСТ»  
по направлениям «Ядро Linux», «Режим безопасных вычислений  
и «Основы Docker»**

**Уважаемые кандидаты!**

Мы рады, что вы выбрали нашу программу. В этом году было принято решение о проведении конкурсного отбора. Ниже представлено 10 задач. Требуется решить все 10 задач нет. Сделайте столько, сколько сможете. Количество — не главный критерий, основным методом отбора будет чистота написанного кода и его работоспособность, комментарии и метод решения поставленной проблемы. Сроки выполнения заданий индивидуальны и выдаются каждому кандидату отдельно. Задания являются общими для всех программ практики.

Решения задач вы можете хранить как на своем github, так и передать их в виде упакованного архива. В случае использования архива — каждое задание должно быть в отдельном каталоге с номером задачи, также в корне каталога задания следует выполнить команду `git init`, чтобы создать локальный git-репозиторий, далее через команду `git add` добавлять файлы и командой `git commit -m «текст»` писать сообщение коммита. Коммитов должно быть несколько, сама по себе ценность единственного коммита с полным решением задачи стремится к нулю. Таким образом мы сможем увидеть ваши коммиты, стиль их написания и атомарность.

Задания следует выполнять на любой GNU/Linux операционной системе (лично мы предпочитаем Debian). Те задания, где есть указание использовать языки C или C++, следует выполнять с использованием компиляторов `gcc` и `g++`. Написание Makefile дает дополнительные баллы в конкурсе, ведь нам нужно будет как-то скомпилировать ваши задачи (или хотя бы файл README.md с командами компиляции, но Makefile ценнее по баллам).

Удачи в выполнении заданий! Решения присылайте в ответном письме на почту.

**P.S. Мы считаем, что умеем отличать код, написанный нейросетью.**

## Задания

1. Написать многопоточную программу на C, использующую библиотеку pthreads, для параллельной сортировки массива любым методом.

2. Реализовать программу на языке C: чат между двумя процессами через именованные каналы (FIFO) или сокеты, используя системные вызовы.

3. Написать программу на C:

Отсортировать строки текстового файла различными способами. У программы 3 аргумента командной строки: входной файл (произвольного размера без ограничений), выходной файл, способ сортировки. Отсортировать входной файл указанным способом, удалить пустые строки без видимых символов и вывести отсортированные данные в выходной файл и на экран.

4. Написать парсер аргументов командной строки на языке C (через getopt и getopt\_long). Программа принимает опции -m -c -s -t, а также --elbrus=[1c+, 2c+, 2c3, 4c, 8c, 16c] и любое количество строк не-опций (без символов - и -- перед ними).

Нужно парсить их в любых комбинациях, например:

```
prg -tm -s --elbrus=16c file0 --elbrus=2c3 file1
```

(Пример вывода): Options are correct: t, m, c, elbrus=16c, elbrus=2c3, non-options: file0, file1.

```
prg -tm -x --elbrus=16c file0
```

(Пример вывода): Options are incorrect.

Доп. задание - вывод некорректной опции на экран.

5. Написать программу на языке C, которая выполняет некоторый системный вызов, например open, и затем выводит возвращаемую ошибку. Использовать в программе разные способы вывода сообщения об ошибке: вывод значения errno, вывод соответствующего сообщения из системного массива сообщений об ошибках sys\_errlist[], вызов библиотечной функции perror.

6. Написать программу на C для получения справочной информации о файлах с помощью системного вызова stat. Имя файла передается в программу из командной

строки. Получить справочную информацию для различных типов файлов (обычный файл, каталог, специальные байт- и блок-ориентированные файлы).

7. Написать программу на C: открыть файл (произвольного размера без ограничений), доступный для чтения. Имя файла передается из командной строки. После открытия файла создать параллельный процесс с помощью `fork`. В обоих процессах создать свои файлы для записи, читать информацию из общего файла и копировать ее в собственные выходные файлы (не на экран). Вывести на экран содержимое полученных файлов из порожденного процесса по окончании записи в файл и из родительского процесса, дождавшись завершения порожденного процесса.

8. Написать программу на C. Программа перемножает квадратные матрицы размером 1000x1000 с 1-секундной задержкой между каждой итерацией умножения. Требуется изменить реакцию на первый полученный сигнал SIGINT в программе: при получении SIGINT вывести на экран текущие итераторы столбца и строки матриц и восстановить реакцию на SIGINT на реакцию по умолчанию. При получении второго сигнала SIGINT программа должна завершиться. Реакцию попытаться изменить как при помощи `signal()`, так и при помощи `sigaction()`.

9. Написать программу на любом языке, реализующую следующий алгоритм: Дан конечный одномерный массив случайных целых чисел. Найти максимальную непрерывную последовательность чисел с максимальной суммой элементов в ней. Непрерывная последовательность означает, что каждое следующее последовательности число больше, чем предыдущее.

10. Разработать парсер логических выражений на языке C или C++, который вычисляет значение переменных. Пример выражения: (A AND B) OR (C AND NOT D)