

федеральное государственное автономное образовательное учреждение высшего образования  
Уральский федеральный университет имени первого Президента России Б.Н. Ельцина

Институт естественных наук и математики  
Кафедра вычислительной математики и компьютерных наук

# ОПЕРАЦИОННЫЕ СИСТЕМЫ

## ПРАКТИЧЕСКИЕ ДОМАШНИЕ ЗАДАНИЯ

## ENCONVERT.CMD

Максимальный балл – 10

### Легенда

Однажды решил Дима заняться наукой. По совету профессора он скачал русскоязычные статьи, опубликованные на сайтах научных конференций. Однако, на этих конференциях в качестве кодировки используют не UTF-8, поэтому вместо текста Дима видит кракозябры. Помогите Диме прочитать статьи.

### Постановка задачи

Дан каталог с русскими текстами в кодировке cp866. Напишите скрипт, который конвертирует все txt-файлы в кодировку utf-8. Структура каталогов должна быть сохранена.

Передаваемые аргументы: имя директории.

[Скачать исходные тексты](#)

### Минимальные требования к выполнению – 5 баллов

[enconvert.cmd](#)

Программа должна вывести либо подробную справку по использованию, либо сообщение о том, что для получения справки требуется запуск с ключом /?. После чего программа должна завершить работу.

[enconvert.cmd /?](#)

Программа должна вывести подробную справку по использованию и завершить работу.

[enconvert.cmd directory](#)

Программа должна обработать все текстовые файлы из папки directory и завершить работу.

Допускается создание рядом с файлом input.txt файла input.txt-utf8.txt.

Корректно обрабатывайте переданные аргументы.

В случае возникновения каких-либо ошибок сообщите об этом пользователю.

### Повышенные требования к выполнению – 10 баллов

Выполните все минимальные требования за исключением допущения к созданию переименованного файла рядом с каждым исходным. После конвертации все новые файлы должны заменить все старые.

При использовании временного файла убедитесь, что он отсутствовал в системе. После завершения работы программы удалите все лишние файлы.

### Подсказки

```
FOR /R %DIR% %%f IN (*.txt) DO echo %%f
```

Может пригодиться iconv.exe из [www.mingw.org](http://www.mingw.org)

# IMAGEMAGICK.SH

## Максимальный балл – 10

### Легенда

Задумал Гоша быть не таким как все и стать фотографом. При этом он хочет ходить не с попсовой камерой, а с Nokia Lumia 1020 – 41 мегапиксель всё-таки! Задумано – сделано. Первая осенняя фотосессия готова. Осталось только увековечить себя, добавив свою подпись на каждую фотографию. Пусть все знают, что эти фотографии сделал Гоша! И, конечно же, нельзя давать изображения в полном размере бесплатно – напечатают календарей, сделают фоном рабочего стола. К тому же говорят, что не комильфо делать веб-страницы размером в полтора гигабайта, хотя на ней всего лишь около 100 прекрасных фотографий, по 14 мегабайт каждая. И напоследок, у каждого уважающего себя фотографа должна быть киллер-фича. Листья непременно должны кружиться – gif не теряют популярности! Помогите Гоше словить хайп.

### Постановка задачи

Произведите манипуляции с утилитой `imagemagick`, выполнив только один из предложенных вариантов.

#### Вариант 1

Дано `jpg`-изображение. Напишите скрипт, который сделает из него анимированное `gif`-изображение. Анимация заключается в повороте `jpg`-изображения по часовой стрелке.

Передаваемые аргументы: угол поворота изображения, время анимации, имя исходного изображения, имя итогового файла.

Если имя итогового файла не передано, то он должен называться «исходное\_имя\_без\_расширения.gif».

#### Вариант 2

Дан каталог с `jpg`-изображениями. Напишите скрипт, который добавит надпись в правый нижний угол всех изображений из каталога.

Передаваемые аргументы: имя исходной директории, текст надписи, имя директории с результатами, [имя шрифта], [размер шрифта].

Итоговые файлы должны называться «исходное\_имя\_без\_расширения\_annotated.jpg».

#### Вариант 3

Дан каталог с `jpg`-изображениями. Напишите скрипт, который изменит размеры всех «больших» `jpg`-изображений в каталоге. Необходимо сделать так, чтобы изображение вместилось в прямоугольник с размерами `width` и `height`. Если какое-либо из измерений больше заданного, то масштабируйте изображение с сохранением пропорций, иначе – оставьте изображение без изменений.

Передаваемые аргументы: имя исходной директории, `width`, `height`, имя итоговой директории.

### Требования к выполнению – 10 баллов

`./imagemagick.sh`

Программа должна вывести либо подробную справку по использованию, либо сообщение о том, что для получения справки требуется запуск с ключом `-h` или `--help`. После чего программа должна завершить работу.

`./imagemagick.sh -h` или `./imagemagick.sh --help`

Программа должна вывести подробную справку по использованию и завершить работу.

Корректно обрабатывайте переданные аргументы.

В случае возникновения каких-либо ошибок сообщите об этом пользователю.

В коде не должно быть ошибок и предупреждений по мнению сервиса [shellcheck.net](https://shellcheck.net).

### Подсказка

Скорее всего пакеты, связанные с `imagemagick`, у вас уже установлены.

`Convert` – основная утилита для преобразований изображений.

`Identify` – утилита для получения информации об изображениях.

## CLEARPATH.SH

Максимальный балл – 15

### Легенда

Купила Вера мощнейший ноутбук, поставила на него современную операционную систему BolgenOS. Спустя время, ноутбук начал нещадно тормозить. Даже эксклюзивно вышедшая под линукс Half Life 3, запускаемая из терминала, открывалась долго и неохотно. Помогите Вере обрести веру в мощнейший ноутбук и почистить систему от всего лишнего. Например, многие программы, которые устанавливались и затем удалялись, оставили после себя в PATH то, что пользователю уже никогда не пригодится.

### Постановка задачи

Напишите скрипт, который проверяет, что во всех директориях из PATH есть исполняемые файлы.

Если исполняемые файлы отсутствуют или проверяемая директория не существует, то скрипт должен вывести очищенный от всего «лишнего» PATH – сам PATH изменять не надо.

Если скрипт запущен с аргументом, отличным от -h и --help, то вместо PATH должна быть выполнена проверка содержимого этого аргумента.

### Минимальные требования к выполнению – 10 баллов

`./clearpath.sh -h` или `./clearpath.sh --help`

Программа должна вывести подробную справку по использованию и завершить работу.

`./clearpath.sh`

Программа должна вывести на экран очищенный от всего «лишнего» PATH и завершить работу.

`./clearpath.sh /bin:/usr/games:/usr/bin`

Программа должна вывести на экран очищенное от всего «лишнего» содержимое переданного аргумента и завершить работу.

Корректно обрабатывайте переданные аргументы.

В случае возникновения каких-либо ошибок сообщите об этом пользователю.

В коде не должно быть ошибок и предупреждений по мнению сервиса [shellcheck.net](https://shellcheck.net).

### Повышенные требования к выполнению – 15 баллов

Выполните все минимальные требования.

Избавьтесь от дублирующихся путей в PATH и будьте внимательней к именам файлов.

# LOGANALYSER.SH

## Максимальный балл – 10

### Легенда

Веб-программист Катя обнаружила очень странные дела, происходящие с сайтом. К счастью, всё, происходящее на сервере, на всякий случай пишется в текстовый файл. Это привело к тому, что файл с логами достиг небывалых размеров и работать с ним даже в Sublime Text стало совсем некомфортно. Что уж говорить об обычном виндовом notepad.exe, который зависал при попытке просто посмотреть логи. К тому же менеджер проекта с продавщиками периодически просят узнать, насколько популярна та или иная страница сайта. Упростите Кате жизнь.

### Постановка задачи

Дан лог некоторого сервера. Напишите скрипт, который решает только одну из предложенных задач.

#### Вариант 1

Найдите медиану по времени обработки запроса для указанного в параметрах source. Если параметр не указан, то поиск осуществляется для всех source.

#### Вариант 2

Найдите 99 перцентиль по времени обработки запроса.

#### Вариант 3

Выведите все записи за последние N минут к заданному моменту времени.

#### Вариант 4

Выведите список уникальных ip, для которых были записи уровня ERROR или FATAL.

#### Вариант 5

Найдите самый популярный ресурс на сервере.

[Скачать файл для обработки](#)

### Формат файла

level | date | ip | source | exec\_time

level	Уровень логирования (INFO, WARN, ERROR, FATAL, DEBUG). Это поле всегда содержит 6 символов. Если level меньше 6, то оставшееся место добивается пробелами.
date	Время запроса в формате yyyy-mm-dd hh:mm:ss.
ip	IP-адрес, с которого пришёл запрос. Это поле всегда содержит 15 символов. Если ip меньше 15, то оставшееся место забивается пробелами.
source	Запрашиваемый ресурс.
exec_time	Время обработки запроса.

### Требования к выполнению – 10 баллов

[./loganalyser.sh -h](#) или [./loganalyser.sh --help](#)

Программа должна вывести справку по использованию и завершить работу.

Во всех вариантах на входе подается имя файла с логами.

Корректно обрабатывайте переданные аргументы.

В случае возникновения каких-либо ошибок сообщите об этом пользователю.

В коде не должно быть ошибок и предупреждений по мнению сервиса [shellcheck.net](#).

# SHARDCOLLECT.SH

Максимальный балл – 15

## Легенда

Решил Антон стать мобильным разработчиком, установил Xamarin, начал программировать. Внезапно начало заканчиваться место на жёстком диске. Xamarin не только в процессе разработки оставлял после себя много лишнего, но и уже установленные обновления никуда автоматически не пропадали. К счастью, Антон читал про утилиту Isof, которая поможет найти, какие файлы уже не используются в системе. Если файлы не используются, значит временные директории с ними после недобросовестной программы можно будет спокойно удалить. Помогите Антону очистить место на жёстком диске.

## Постановка задачи

Используя утилиту Isof, найдите в заданной директории все подкаталоги первого уровня, в которых не открыт ни один файл. Найденные директории выведите по одной в строке. Если не удалось ничего найти, то ничего не выводите и установите код возврата программы, отличный от нуля. Запуск программы без параметров означает обработку текущей директории.

## Минимальные требования к выполнению – 10 баллов

`./shardcollect.sh -h` или `./shardcollect.sh --help`

Программа должна вывести справку по использованию и завершить работу.

`./shardcollect.sh directory`

Программа должна найти и вывести все подкаталоги первого уровня директории `directory`, в которых не открыт ни один файл.

`./shardcollect.sh`

Программа должна найти и вывести все подкаталоги первого уровня текущей директории, в которых не открыт ни один файл.

Корректно обрабатывайте переданные аргументы.

В случае возникновения каких-либо ошибок сообщите об этом пользователю.

В коде не должно быть ошибок и предупреждений по мнению сервиса [shellcheck.net](https://shellcheck.net).

## Повышенные требования к выполнению – 15 баллов

Утилита Isof запускается только один раз.

## Подсказки

Для обработки аргументов программы можно использовать команды `case` и `shift`.

Обратите внимание, что одновременно с директорией `Desktop` рядом может существовать директория `Desktop1`.

## NORMCONF.SH

Максимальный балл – 20

### Легенда

Ох уж эти конфигурационные файлы... Хочешь сказать программе, чтобы она делала что-то раз в год? Считай количество секунд в этом году. Или смотришь на огромное число и не понимаешь, что это – месяц, неделя или другая астрономическая единица. Куда удобнее писать значения «2 года 2 секунды». И каждый без калькулятора и бумажки сразу поймет, что к чему. Вот и решил Галлям разобраться с этой проблемой и сделать так, чтобы все писали конфиги именно в таком формате. Но не переписывать ведь существующие программы для такой обработки? Поэтому Галлям решил размять руки и написать на любимом `bash` программу, которая переводила бы конфигурационные файлы из человекопонятного формата в секунды, граммы и прочие стандартные единицы. Помогите Галляму сделать жизнь программистов проще.

### Постановка задачи

Дан конфигурационный файл в следующем формате (слева от знака могут быть любые английские буквы):

```
time=4s
distance=8m
longtime=15h
weight=42g
```

Допустимые единицы измерения в «ненормализованном» конфигурационном файле: время – s, min, h, d; расстояние – mm, sm, dm, m, km; вес – mg, g, kg, t.

Напишите скрипт, который формирует «нормализованный» конфигурационный файл. Все единицы должны быть приведены к одному измерению: время – в секундах, расстояние – в метрах, вес – в килограммах.

Передаваемые аргументы: имя конфигурационного файла.

### Минимальные требования к выполнению – 10 баллов

`./normconf.sh`

Программа должна вывести либо подробную справку по использованию, либо сообщение о том, что для получения справки требуется запуск с ключом `-h` или `--help`. После чего программа должна завершить работу.

`./normconf.sh -h` или `./normconf.sh --help`

Программа должна вывести подробную справку по использованию и завершить работу.

`./normconf.sh config.txt`

Программа должна сформировать «нормализованный» конфигурационный файл и завершить работу.

Итоговый конфигурационный файл должен называться «исходное\_имя\_без\_расширения\_normal.txt».

Корректно обрабатывайте переданные аргументы.

В случае возникновения каких-либо ошибок сообщите об этом пользователю.

В коде не должно быть ошибок и предупреждений по мнению сервиса [shellcheck.net](https://shellcheck.net).

### Повышенные требования к выполнению – 15 баллов

Выполните все минимальные требования.

Обработайте следующие ситуации:

```
time=4h 2s
weight=5kg - 700g + 40mg
```

Если в какой-либо строке конфигурационного файла обнаружись несочетаемые данные, выведите информацию об ошибке с номером строки и именем некорректного параметра.

Поддерживаемые арифметические операции – сложение и вычитание. Если операция не указана, то единицы измерения нужно складывать.