

Практикум на ЭВМ

Семестровая работа №1.

bash

Баев А.Ж.

Казахстанский филиал МГУ

08 октября 2018

Семестровая работа.

1. Менеджер тестирования judge.

- компилирует программу
- запускает тестирующую программу на подготовленном наборе тестов
- подводит итог

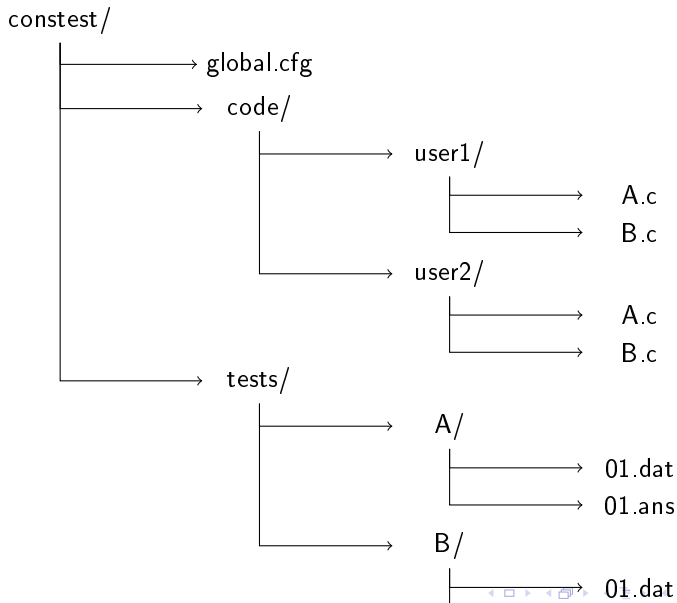
2. Тестирующая программа test.

- запускает пользовательскую программу на указанном наборе входных тестов
- вывод стандартного потока сравнивает с набором выходных тестов с помощью указанного чекера

3. Чекеры.

- checker_byte сверяет на соответствие вывод программы и правильный ответ (байт в байт).
- checker_int сверяет на соответствие вывод программы и правильный ответ (целое число, игнорируя пробельные символы).

Структура



Пример запуска

Протестировать всё

```
./judge contest
```

Результаты выводятся в стандартный поток и дублируются в файл results.csv.

```
user  A B Sum
user1 + +   2
user2 + -   1
```

Пример запуска

Протестировать одну программу

```
./test contest/user2/A.c contest/tests/A checker_int
```

Результаты выводятся в стандартный поток и дублируются в файл results.txt.

```
-x+-+
```

- — неверный ответ + — верный ответ x — ошибка запуска (нет файла тестирования или пользовательская программа упала)

Файлы настройки

Файл настройки `global.cfg`:

```
users=2  
problems=2  
score=sum
```

Файл настройки `problem.cfg`:

```
name=A  
tests=5  
checker=byte
```

Файлы настройки — альтернатива

Файл настройки global.xml:

```
<users>
2
</users>
<problems>
2
</problems>
```

Файл настройки problem.json:

```
{
  "tests": "2",
  "checker": "byte"
}
```

Все действия описываем в лог-файл.

```
start test
  user: user1
  problem: B
  tested: 5
  failed: 0
stop test
```

failed — не смог открыть файл с тестом.


```
user, A, B, Sum  
user1, +, +, 2  
user2, +, -, 1
```

Репозиторий на сайте.

```
https://github.com/
```

Клиент

1. Установка и настройка

```
sudo apt install git
```

```
git config --global user.name "valera"  
git config --global user.email "valera@msu.kz"  
git config --global core.autocrlf input  
git config --global core.safecrlf true
```

2. Инициализируем директорию для Git репозитория. Это нужно сделать только один раз для каждого проекта.

```
git init
```

3. Связываем папку с удаленным репозиторием

```
git remote add origin https://github.com/valera/judge
```

1. Добавляем все новые и измененные файлы к контролю

```
git add .
```

2. Помечаем все новые и измененные файлы сообщением (commit)

```
git commit -m "message"
```

- вместо message вписываем сообщение, например "add 2 bugs".

3. Закачиваем код на удаленный репозиторий

```
git push -u origin master
```

1. Скачиваем код с удаленного репозитория

```
git pull origin master
```

Закачиваем код на удаленный репозиторий в первый раз с указанием ветки

```
git push -u origin master
```

Дополнительные настройки

```
git config --global push.default simple  
git push --set-upstream origin master
```

Далее без ветки

```
git push
```

[http://rgblog.ru/page/
git-dlja-chajnika-komandy-kotorye-pomogut-nachat-rabotu](http://rgblog.ru/page/git-dlja-chajnika-komandy-kotorye-pomogut-nachat-rabotu)