

# Технологии разработки программного обеспечения

2017 / 2018, 1 курс, 2 семестр

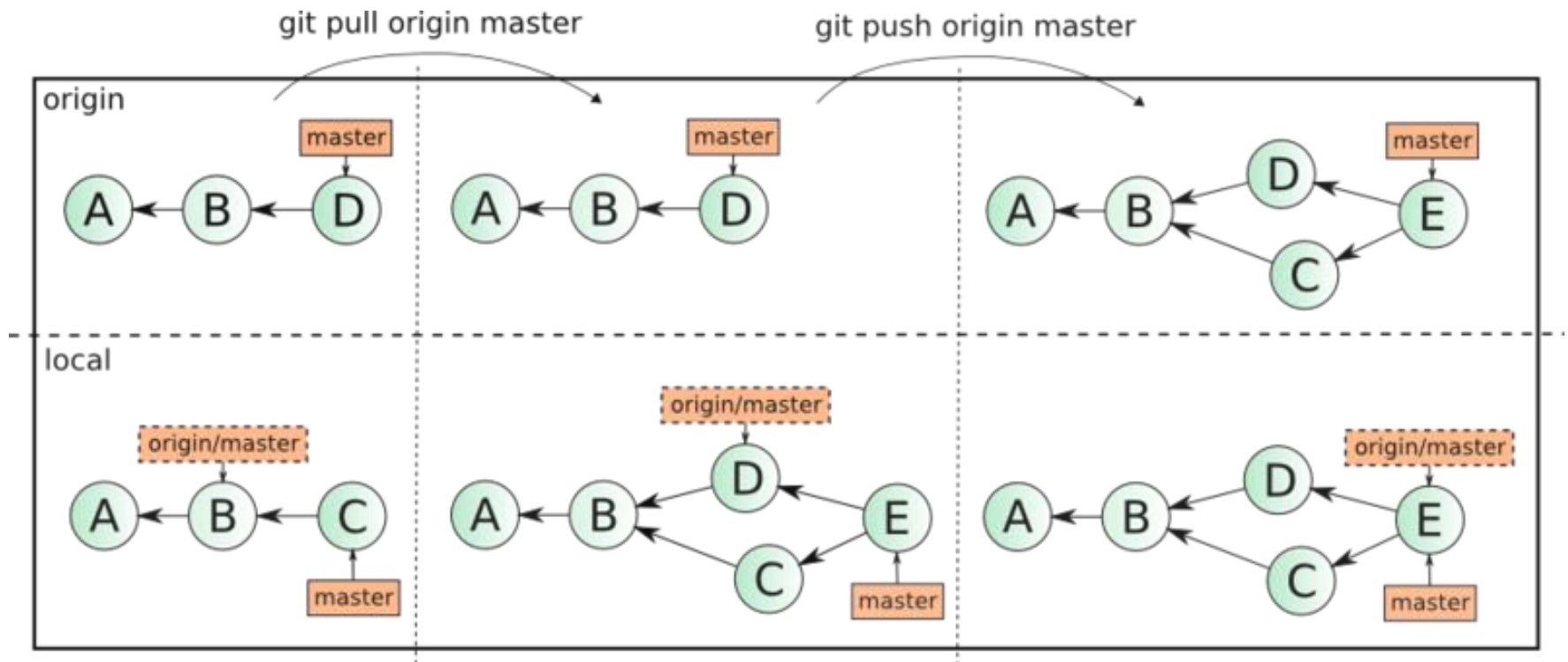
Пудов Сергей Григорьевич

# Лекция 6

---

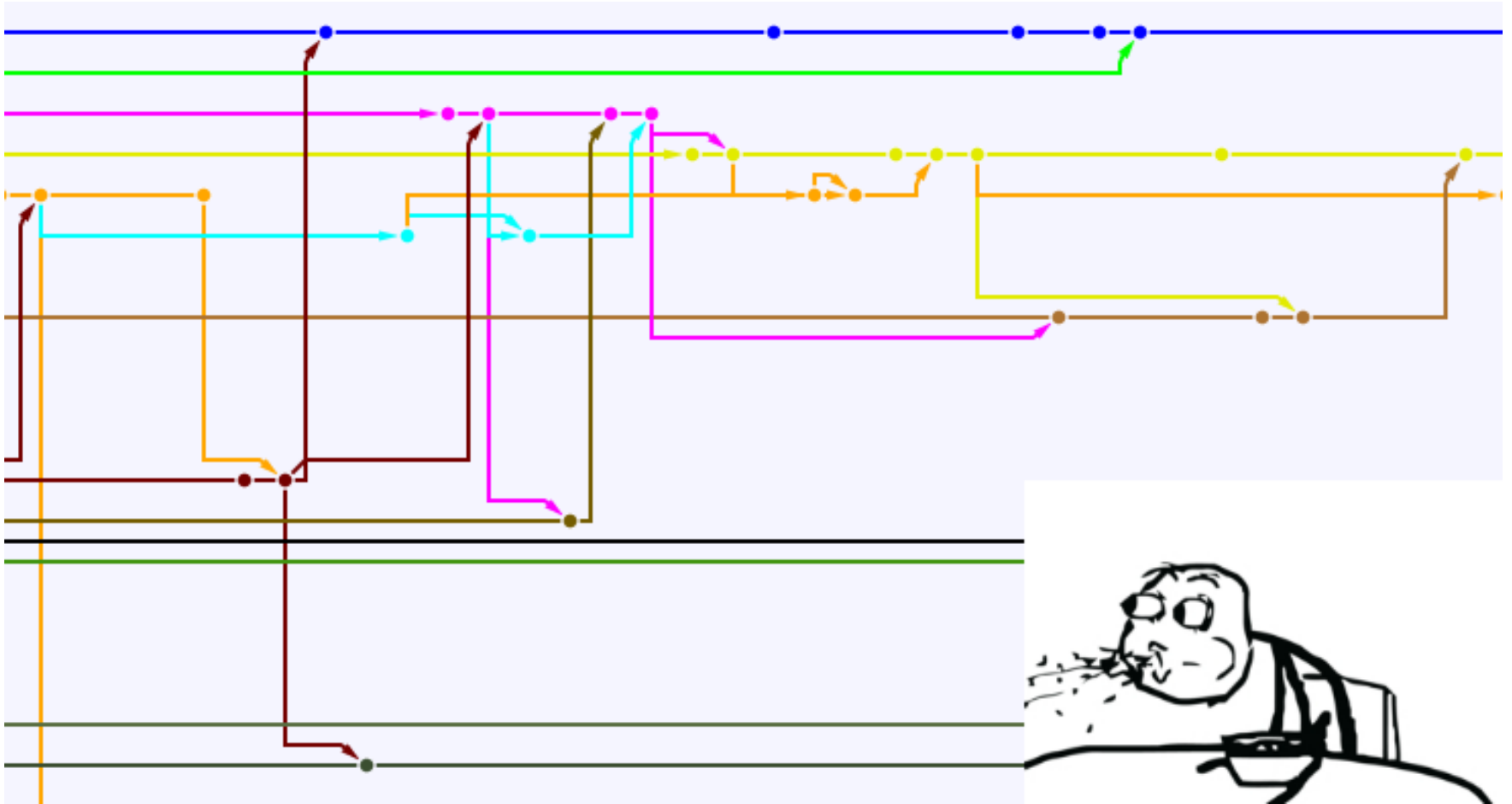
- ▶ `git pull --rebase`
- ▶ Многофайловые приложения
  - ▶ Интерфейс функции
  - ▶ Include guards
- ▶ Makefile

# git pull --rebase

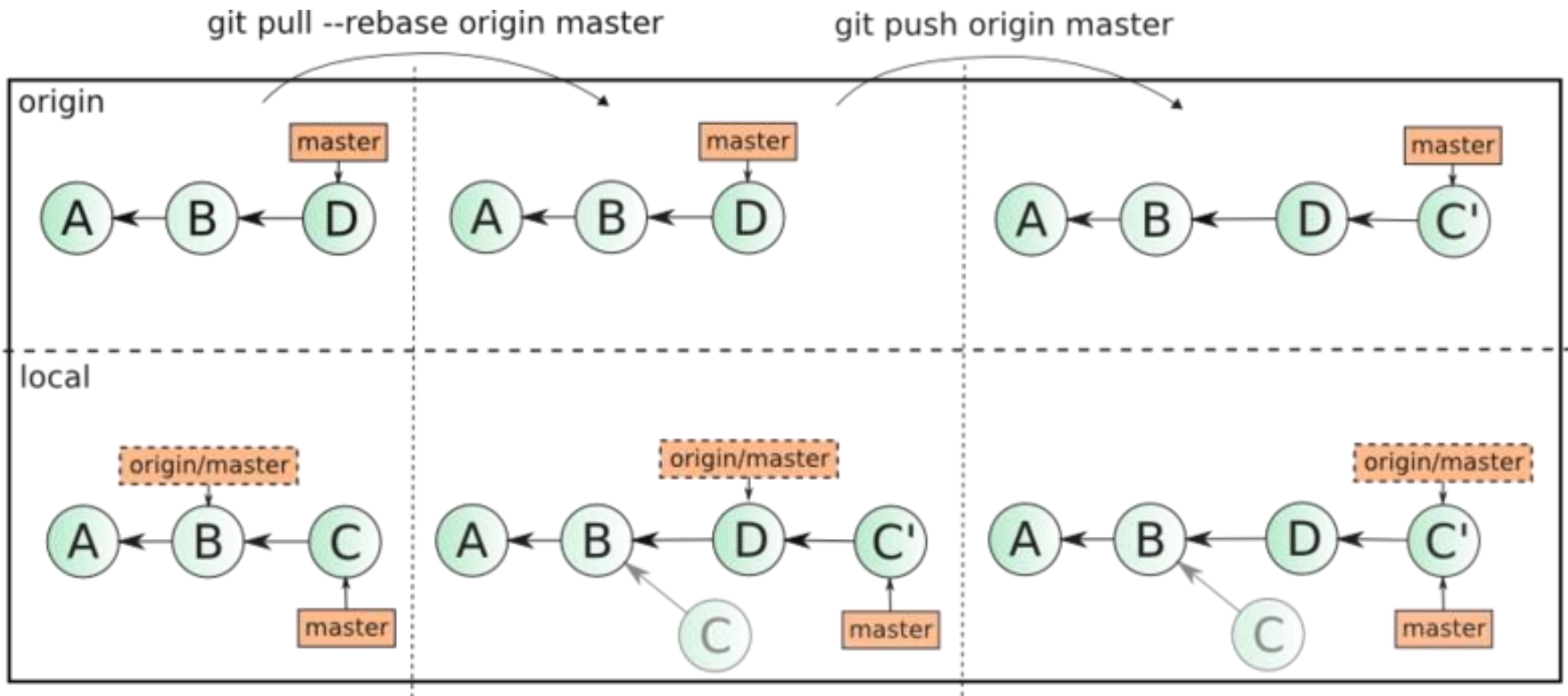


<https://habrahabr.ru/post/161009/>

# git pull --rebase



# git pull --rebase



# Многофайловые приложения

---

▶ //////////////// Prog.c :

```
int foo( double d )
{
    if ( d > 5.0 )
        return 2;
    return 1;
}

int main()
{
    double val = 10;
    int res = foo( val );
    printf( "res = %d\n", res );
}
```

# Многофайловые приложения

---

▶ // foo.c :

```
int foo( double d )
{
    if ( d > 5.0 )
        return 2;
    return 1;
}
```

▶ // Prog.c :

```
int main()
{
    double val = 10;
    int res = foo( val );           // вызов неизвестной функции !!!
    printf( "res = %d\n", res );
}
```

# Многофайловые приложения

---

▶ *//////////* foo.h :

```
#ifndef _MYPROJECT_FOO_H
#define _MYPROJECT_FOO_H
int foo( double d ); // интерфейс функции
#endif
```

▶ *//////////* Prog.c :

```
#include "foo.h"
int main()
{
    double val = 10;
    int res = foo( val );
    printf( "res = %d\n", res );
}
```



# Makefile

---

цель : зависимости

[tab] команды

```
$ gcc main.c foo.c -o prog.exe
```

```
// Makefile
```

```
all:
```

```
    gcc main.c foo.c -o prog.exe
```

# Makefile #1

---

all: prog

prog: main.o foo.o

gcc main.o foo.o -o prog.exe

main.o : main.c

gcc -c main.c

foo.o : foo.c

gcc -c foo.c

clean:

rm -rf \*.o \*.exe

# Makefile #2

---

CC = gcc

CFLAGS = -c -Wall

main.o : main.c

\$(CC) \$(CFLAGS) main.c

foo.o : foo.c

\$(CC) \$(CFLAGS) foo.c

# Makefile #3

---

CC = gcc

CFLAGS = -c -Wall

EXECUTABLE = prog

SOURCES = main.c foo.c

OBJECTS=\$(SOURCES:.c=.o)

all: \$(SOURCES) \$(EXECUTABLE)

\$(EXECUTABLE) : \$(OBJECTS)

\$(CC) \$(OBJECTS) -o \$@

.c.o :

\$(CC) \$(CFLAGS) \$< -o \$@

clean:

rm -rf \*.o \*.exe

# Фиктивные цели

---

- ▶ all — является стандартной целью по умолчанию. При вызове make ее можно явно не указывать.
- ▶ clean — очистить каталог от всех файлов полученных в результате компиляции.
- ▶ install — произвести инсталляцию
- ▶ uninstall — и деинсталляцию соответственно.

Чтобы make не искал файлы с такими именами, их следует определить при помощи директивы .PHONY.

.PHONY: all clean install uninstall

# Продолжение в следующей лекции...

---



# ТЗ к проекту: пример

---

- ▶ **О проекте**
- ▶ **Игровая механика**
  - ▶ **Правила**
    - ▶ Действия, за которые счет увеличивается:
    - ▶ Действия, за которые счет уменьшается:
    - ▶ Условия завершения игры:
- ▶ **Интерфейс приложения**
  - ▶ **Главное меню**
  - ▶ **Игровой интерфейс**
- ▶ **Данные извне**
  - ▶ ...

## ТЗ к проекту: данные извне

---

### ► Данные извне

**words.txt.** Содержит предложения, доступные для вывода в приложении. Каждое предложение начинается с новой строки; слова и запятые отделяются пробелом. **Пример:**

*Первая строка:* И слово , следующее слово

*Вторая строка:* Слово слово

**results.txt.** Содержит информацию о лучших результатах. Всего в файле пять строк, оформленных **по следующему шаблону:**

<Счет> <Имя латинскими символами>

100500 Vasya

228 Ivan



# ТЗ к проекту: правила

---

Весь игровой процесс завязан на **количестве очков**. При запуске **новой игры**, счетчик очков принимает значение равное нулю ( $score = 0$ ). Если текущий счет пользователя в любой момент времени будет меньше нуля ( $score < 0$ ), **игра завершается**. Счет изменяется в зависимости от действий игрока.

## **Действия, за которые счет увеличивается:**

Правильный ввод буквы (+25)

Ввод слова без единой ошибки при наборе (+50)

Ввод предложения без единой ошибки при наборе (+100)

## **Действия, за которые счет уменьшается:**

Неправильный ввод буквы (-15)

Истечение времени, выделенного на ввод буквы (-25)

## **Условия завершения игры:**

Счет игрока отрицателен.

Был закончен ввод предложения при максимальном темпе игры. *(Как говорилось выше, время, выделяемое на ввод слова, постепенно уменьшается до определенного предела)*. При достижении этого условия, счет игрока сохраняется в таблицу лидеров по определенным правилам (см. ниже)