

# Поиск ошибок и причин изменений в коде

Выполнила студентка группы ПриН-466  
Никифорова Анна

# Команда hg bisect

Примеры ситуаций, когда полезно использовать команду hg bisect:

- Самая последняя версия программного обеспечения имеет ошибку, и Вы помните, что ее не было несколько недель назад, но не знаете, когда она появилась.
- Вы исправили ошибку в спешке, и теперь пришло время закрыть запись об ошибке в багтрекере вашей команды. Багтрекер данных требует ID ревизии, когда вы закрываете записи, но Вы не помните, в какой ревизии исправили ошибку.

# Команда hg bisect

Команда hg bisect работает по шагам:

1. Вы запускаете Ваш бинарный тест  
Если тест успешен, запускаем команду **hg bisect --good**.  
Если неуспешен, запускаете команду **hg bisect --bad**
2. Mercurial использует вашу информацию, чтобы решить, какая ревизия для тестирования следующая
3. Он обновляет рабочий каталог до этой ревизии и процесс повторяется сначала до тех пор пока hg bisect не идентифицирует уникальный набор изменений, который знаменует собой точку, где Ваш тест перешел из "успешного" в "неуспешный"

Опции:

- |               |                                     |
|---------------|-------------------------------------|
| -r (--reset)  | сбросить bisect.                    |
| -g (--good)   | отметить ревизию как "хорошую"      |
| -b (--bad)    | отметить ревизию как "плохую"       |
| -s (--skip)   | пропустить тестовый набор изменений |
| -e (--extend) | расширить диапазон bisect           |

# Команда hg bisect

```
1 hg bisect --reset
2 hg bisect --bad
3 hg bisect --good 10
4 Testing changeset 22:e9e43d57c12e (24 changesets remaining, ~4 tests)
5 0 files updated, 0 files merged, 12 files removed, 0 files unresolved
6 [ваше тестирование]
7 hg bisect --bad
8 Testing changeset 16:a20d4936611f (12 changesets remaining, ~3 tests)
9 0 files updated, 0 files merged, 6 files removed, 0 files unresolved
10 [ваше тестирование]
11 hg bisect --good
12 ...
13 hg bisect --good
14 The first bad revision is:
15 changeset: 22:e9e43d57c12e
16 user:      Bryan O'Sullivan <bos@serpentine.com>
17 date:      Thu Feb 02 14:09:26 2012 +0000
18 summary:   buggy changeset
19 hg bisect --reset
```

# Команда hg log

Допустим, Вам нужно полное описание изменений или список измененных файлов, чтобы узнать, та ли это ревизия, что Вам нужна.

Команда `hg log` с ключом `-v` (`--verbose`) предоставляет такую возможность.

Пример:

```
$ hg log -v -r 3
changeset: 3:0272e0d5a517
user: Bryan O'Sullivan
date: Sat Aug 16 22:08:02 2008 +0200
files: Makefile
description:
Get make to generate the final binary from a .o file.
```

# Команда hg log

Или ситуация, когда вы хотите видеть и описание, и то как изменялось содержимое. Тогда нужен ключ `-p` (`--patch`)

Пример:

```
$ hg log -v -p -r 2
changeset: 2:fef857204a0c
user: Bryan O'Sullivan
date: Sat Aug 16 22:05:04 2008 +0200
files: hello.c
description:
Introduce a typo into hello.c.
```

```
diff -r 82e55d328c8c -r fef857204a0c hello.c
--- a/hello.c Fri Aug 26 01:21:28 2005 -0700
+++ b/hello.c Sat Aug 16 22:05:04 2008 +0200
@@ -11,6 +11,6 @@
int main(int argc, char **argv)
{
- printf("hello, world!\n");
+ printf("hello, world!\");
  return 0;
}
```

# Команда hg annotate

Допустим, Вы знаете ревизию, в которой допущена ошибка. И хотите узнать, чей код привёл к ней (кто автор).

Пример:

```
$hg annotate -unl aFile
```



```
jim 1519:477: a = 4  
bob 1518:468: b = 5  
max 1496:402: return a
```

Опции:

- u (--user)            выводит автора
- n (--number )        выводит номер ревизии
- l (--line-number )    выводит номер строки
- d (--date)            выводит дату

Спасибо за внимание