Application of Git

CAT&DOG Git study 2013
Day 3

Eon Jeong

March 31, 2013

Contents

Contents

Applying patches and forks

Maintaining a repository

Other features

Thanks

Section 1

Applying patches and forks

How to compare?

```c
#include <stdio.h>

int main() {
    const char format[11] = "Hello %s!\n";
    char buf[80] = "World";
    printf(format, buf);
    return 0;
}

```

```c
#include <stdio.h>
#include <time.h>

int main() {
    const char format[11] = "Hello %s!\n";
    char buf[80];
    time_t now = time(NULL);
    strftime(buf, 20, "%F %T", localtime(&now));
    printf(format, buf);
    return 0;
}

```

Diff

```
| 1 |  1 | #include <stdio.h>
|   |  2 | #include <time.h>
| 2 |  3 |
| 3 |  4 | int main() {
| 4 |  5 |     const char format[11] = "Hello %s!\n";
| 5 |    |     char buf[80] = "World";
|   |  6 |     char buf[80];
|   |  7 |     time_t now = time(NULL);
|   |  8 |     strftime(buf, 20, "%F %T", localtime(&now));
| 6 |  9 |     printf(format, buf);
| 7 | 10 |     return 0;
| 9 | 11 | }
| 9 | 12 |
```

diff

Show line-by-line difference (mininum edit) between commits or workspace.

diff

Show line-by-line difference (mininum edit) between commits or workspace.

- Compare file(s) between commits. (2 or *more*)
- - and +, not <and >.
- Display revision ID and adjacent common lines.

`diff`

Show line-by-line difference (mininum edit) between commits or workspace.

- Compare file(s) between commits. (2 or *more*)
- - and +, not <and >.
- Display revision ID and adjacent common lines.

We can use `format-patch` to generate patch file from diff.

remote

Manage remote tracked repositories.

- remote add
  remote rename
  remote remove
- remote set-head
  remote set-branches
  remote set-url
- remote prune
  remote update

merge

Join two or more development histories together.

- Use history from their past common revision, if possible.

fetch

Think about pull without merge. Just update references and objects.

Section 2

Maintaining a repository

branch

List, create, or delete codelines.

tag

List, create, delete or verify a tag object, GPG-signed.

`rebase`

Forward commits to updated stream baseline by differences.

revert

Revert some commits without stripping them.

Section 3

Other features

reset

Transit HEAD to the specified state.

stash

Temporarily store current changes and clean workspace.

config

Handle options, for repository or global(user-wide).

.gitignore

We can specify untracked files by name specs, e.g. for temporary files.

- Wildcard(*) can be used.
- Without /, pplied recursively.
- Files already staged are not affected.

.gitmodules

Some workspace is required to contain other repository *module* inside, e.g. for library use. cf. `submodule`

Thank you.

1. CAT
2. DOG

Thank you.

1. CAT
2. DOG
3. ???
4. PROFIT!