

Patch operation in Git

The patch is a text file, its content is similar to git diff, but in parallel with the code, it also has metadata about commits such as commit IDs, dates, commit messages ... We can Create a patch from commits and others can apply them to their repository.

The patch is a text file, its content is similar to git diff, but in parallel with the code, it also has metadata about commits such as commit IDs, dates, commit messages . We can Create a patch from commits and others can apply them to their repository.

Jerry performs the strcat function for his project. Jerry can create a patch of his code and send it to Tom. After that, he can apply the received patch to his code.

Jerry uses the git format-patch command to create a patch for the latest commits. If you want to create a patch for a specific commit, then use COMMIT_ID with the format-patch command.

```
[jerry @ CentOS project] $ pwd
/ home / jerry / jerry_repo / project / src

[jerry @ CentOS src] $ git status -s
M string_operations.c
?? string_operations

[jerry @ CentOS src] $ git add string_operations.c

[jerry @ CentOS src] $ git commit -m "Added my_strcat function"

[master b4c7f09] Added my_strcat function
1 files changed, 13 insertions (+), 0 deletions (-)

[jerry @ CentOS src] $ git format-patch -1
0001-Added-my_strcat-function.patch
```

The above command creates the .patch files inside the current working directory. Tom can use the patch to edit his file. Git provides two commands to apply patches that are git am and git apply, in a separate way. Git applies to editing internal files without creating commits, while git am modifying the file and also creating commits.

To apply patch and commit, use the following command:

```
[tom @ CentOS src] $ pwd
/ home / tom / top_repo / project / src

[tom @ CentOS src] $ git diff

[tom @ CentOS src] $ git status -s
```

```
[tom @ CentOS src] $ git apply 0001-Added-my_strcat-function.patch
```

```
[tom @ CentOS src] $ git status -s
M string_operations.c
?? 0001-Added-my_strcat-function.patch
```

This patch has been successfully applied, now we can observe the modifications using the git diff command.

```
[tom @ CentOS src] $ git diff
```

The above command produces the following result:

```
diff --git a / src / string_operations.cb / src / string_operations.c
index 8ab7f42.f282fcf 100644
--- a / src / string_operations.c
+++ b / src / string_operations.c
@@ -1,5 +1,16 @@
#include
+ char * my_strcat (char * t, char * s)
diff --git a / src / string_operations.cb / src / string_operations.c
index 8ab7f42.f282fcf 100644
--- a / src / string_operations.c
+++ b / src / string_operations.c
@@ -1,5 +1,16 @@
#include
+ char * my_strcat (char * t, char * s)
+
+ {
+
+ char * p = t;
+
+
+
+ while (* p)
+ ++ p;
+
+ while (* p ++ = * s ++)
+ ;
+ return t;
+
+ }
+
+ size_t my_strlen (const char * s)
+ {
+ const char * p = s;
@@ -23,6 +34,7 @@ int main (void)
{
```

According to Tutorialspoint

Previous post: Tag operation in Git

Next lesson: Managing branches in Git

You finished reading the article "**Patch operation in Git**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides.

Thank you for reading and for following us regularly.
