Git - Update Operation

Modify Existing Function

Tom performs the clone operation and finds a new file string.c. He wants to know who added this file to the repository and for what purpose, so, he executes the **git log** command.

```
[tom@CentOS ~]$ git clone gituser@git.server.com:project.git
```

The above command will produce the following result –

```
Initialized empty Git repository in /home/tom/project/.git/
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
Receiving objects: 100% (6/6), 726 bytes, done.
remote: Total 6 (delta 0), reused 0 (delta 0)
```

The Clone operation will create a new directory inside the current working directory. He changes the directory to newly created directory and executes the **git log** command.

```
[tom@CentOS ~]$ cd project/

[tom@CentOS project]$ git log
```

The above command will produce the following result –

```
commit d1e19d316224cddc437e3ed34ec3c931ad803958
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 08:05:26 2013 +0530

Changed return type of my_strlen to size_t


commit 19ae20683fc460db7d127cf201a1429523b0e319
Author: Tom Cat <tom@tutorialspoint.com>
Date: Wed Sep 11 07:32:56 2013 +0530
```

```
Initial commit
```

After observing the log, he realizes that the file string.c was added by Jerry to implement basic string operations. He is curious about Jerry's code. So he opens string.c in text editor and immediately finds a bug. In my_strlen function, Jerry is not using a constant pointer. So, he decides to modify Jerry's code. After modification, the code looks as follows –

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

The above command will produce the following result –

```
diff --git a/string.c b/string.c
index 7da2992..32489eb 100644
--- a/string.c
+++ b/string.c
@@ -1,8 +1,8 @@
#include <stdio.h>
-size_t my_strlen(char *s)
+size_t my_strlen(const char *s)
{
  - char *p = s;
  + const char *p = s;
  while (*p)
  ++p;
}
```

After testing, he commits his change.

```
[tom@CentOS project]$ git status -s
M string.c
?? string

[tom@CentOS project]$ git add string.c

[tom@CentOS project]$ git commit -m 'Changed char pointer to const char pointer'
[master cea2c00] Changed char pointer to const char pointer
1 files changed, 2 insertions(+), 2 deletions(-)

[tom@CentOS project]$ git log
```

The above command will produce the following result –

```
commit cea2c000f53ba99508c5959e3e12fff493b
Author: Tom Cat <tom@tutorialspoint.com>
Date: Wed Sep 11 08:32:07 2013 +0530

Changed char pointer to const char pointer


commit d1e19d316224cddc437e3ed34ec3c931ad803958
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 08:05:26 2013 +0530

Changed return type of my_strlen to size_t


commit 19ae20683fc460db7d127cf201a1429523b0e319
Author: Tom Cat <tom@tutorialspoint.com>
Date: Wed Sep 11 07:32:56 2013 +0530
Initial commit
```

Tom uses git push command to push his changes.

```
[tom@CentOS project]$ git push origin master
```

The above command will produce the following result –

```
Counting objects: 5, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 336 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
To gituser@git.server.com:project.git
d1e19d3..cea2c00 master -> master
```

Add New Function

Meanwhile, Jerry decides to implement **string compare** functionality. So he modifies string.c. After modification, the file looks as follows –

```
[jerry@CentOS project]$ git diff
```

The above command will produce the following result –

```
index 7da2992..bc864ed 100644
--- a/string.c
+++ b/string.c
30Git Tutorials
@@ -9,9 +9,20 @@ size_t my_strlen(char *s)
return (p -s );
}
+char *my_strcpy(char *t, char *s)
+
{
  +
  char *p = t;
  +
  + while (*t++ = *s++)
  + ;
  +
  +
  return p;
  +
}
+
int main(void)
{
  int i;
  +
  char p1[32];
  char *s[] =
  {
    "Git tutorials",
    "Tutorials Point"
    @@ -20,5 +31,7 @@ int main(void)
    for (i = 0; i < 2; ++i)
    printf("string lenght of %s = %lu\n", s[i], my_strlen(s[i]));
    +
    printf("%s\n", my_strcpy(p1, "Hello, World !!!"));
    +
    return 0;
```

```
  }
}
```

After testing, he is ready to push his change.

```
[jerry@CentOS project]$ git status -s
M string.c
?? string

[jerry@CentOS project]$ git add string.c

[jerry@CentOS project]$ git commit -m "Added my_strcpy function"
[master e944e5a] Added my_strcpy function
1 files changed, 13 insertions(+), 0 deletions(-)
```

Before push operation, he verifies commit by viewing log messages.

```
[jerry@CentOS project]$ git log
```

The above command will produce the following result –

```
commit e944e5aab74b26e7447d3281b225309e4e59efcd
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 08:41:42 2013 +0530

Added my_strcpy function


commit d1e19d316224cddc437e3ed34ec3c931ad803958
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 08:05:26 2013 +0530

Changed return type of my_strlen to size_t


commit 19ae20683fc460db7d127cf201a1429523b0e319
Author: Tom Cat <tom@tutorialspoint.com>
Date: Wed Sep 11 07:32:56 2013 +0530

Initial commit
```

Jerry is happy with the changes and he wants to push his changes.

```
[jerry@CentOS project]$ git push origin master
```

The above command will produce the following result –

```
To gituser@git.server.com:project.git
! [rejected]
master –> master (non-fast-forward)
error: failed to push some refs to 'gituser@git.server.com:project.git'
To prevent you from losing history, non-fast-forward updates were rejected
Merge the remote changes before pushing again. See the 'Note about
fast-forwards' section of 'git push --help' for details.
```

But Git is not allowing Jerry to push his changes. Because Git identified that remote repository and Jerry's local repository are not in sync. Because of this, he can lose the history of the project. To avoid this mess, Git failed this operation. Now, Jerry has to first update the local repository and only thereafter, he can push his own changes.

Explore our latest online courses and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

Fetch Latest Changes

Jerry executes the git pull command to synchronize his local repository with the remote one.

```
[jerry@CentOS project]$ git pull
```

The above command will produce the following result –

```
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From git.server.com:project
d1e19d3..cea2c00 master –> origin/master
First, rewinding head to replay your work on top of it...
Applying: Added my_strcpy function
```

After pull operation, Jerry checks the log messages and finds the details of Tom's commit with commit ID **cea2c000f53ba99508c5959e3e12fff493ba6f69**

```
[jerry@CentOS project]$ git log
```

The above command will produce the following result –

```
commit e86f0621c2a3f68190bba633a9fe6c57c94f8e4f
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 08:41:42 2013 +0530

Added my_strcpy function


commit cea2c000f53ba99508c5959e3e12fff493ba6f69
Author: Tom Cat <tom@tutorialspoint.com>
Date: Wed Sep 11 08:32:07 2013 +0530

Changed char pointer to const char pointer


commit d1e19d316224cddc437e3ed34ec3c931ad803958
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 08:05:26 2013 +0530

Changed return type of my_strlen to size_t


commit 19ae20683fc460db7d127cf201a1429523b0e319
Author: Tom Cat <tom@tutorialspoint.com>
Date: Wed Sep 11 07:32:56 2013 +0530
Initial commit
```

Now, Jerry's local repository is fully synchronized with the remote repository. So he can safely push his changes.

```
[jerry@CentOS project]$ git push origin master
```

The above command will produce the following result –

```
Counting objects: 5, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 455 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
To gituser@git.server.com:project.git
cea2c00..e86f062 master −> master
```