

old.txt为原文件，new.txt 为已修改后的文件，现在要做的是：制作patch，更新old.txt文件

(1) 查看文件内容

```
[wln@localhost 02]$ cat old.txt
```

123456

qwert

dasfg

fdsaf

```
[wln@localhost 02]$ cat new.txt
```

1123456

qqwert

dasfg

dsaf

(2) 制作patch

说明：老的文件 放在前面

```
[wln@localhost 02]$ diff -Naur old.txt new.txt > a.patch
```

```
[wln@localhost 02]$ cat a.patch
```

--- old.txt 2014-06-09 15:35:34.000000000 +0800

+++ new.txt 2014-06-09 15:35:34.000000000 +0800

@@ -1,4 +1,4 @@

-123456

-qwert

+1123456

+qqwert

dasfg

-fdsaf

+dsaf

(3) diff参数解读

-N 选项确保补丁文件将正确地处理已经创建或删除文件的情况

-a 将所有文件都当做文本文件

-u 选项以统一格式创建补丁文件，这种格式比缺省格式更紧凑些

-r 是一个递归选项，设置了这个选项，diff会将两个不同版本源代码目录中的所有对应文件全部都进行一次比较，包括子目录文件

(4) 执行patch

```
[wln@localhost 02]$ patch -p0 < a.patch
```

```
patching file old.txt
```

这里 -p0表示执行patch所在的路径就是制作patch所在的路径。如果为-p1表示执行patch所在的路径为制作patch所在路径的子目录下，如果为 -p2 则为子目录的子目录下。

所以最好是在哪里制作在哪里执行patch（用-p0）

```
[wln@localhost 02]$ cat old.txt
```

```
1123456
```

```
qqwert
```

```
dasfg
```

```
dsaf
```

```
[wln@localhost 02]$ cat new.txt
```

```
1123456
```

```
qqwert
```

```
dasfg
```

```
dsaf
```

(5) 打好patch，发现问题，想返回，则执行-R（补丁的卸载）

```
[wln@localhost 02]$ patch -R -p0 < a.patch
```

```
patching file old.txt
```

```
[wln@localhost 02]$ cat old.txt
```

```
123456
```

```
qwert
```

```
dasfg
```

```
fdsaf
```

```
[wln@localhost 02]$ cat new.txt
```

```
1123456
```

```
qqwert
```

```
dasfg
```

```
dsaf
```