SVN - BRANCHING

 Copyright © tutorialspoint.com

Branch operation creates another line of development. It is useful when someone wants the development process to fork off into two different directions. Let us suppose you have released a product of version 1.0, you might want to create new branch so that development of 2.0 can be kept separate from 1.0 bug fixes.

In this section, we will see how to create, traverse and merge branch. Jerry is not happy because of the conflict, so he decides to create a new private branch.

```
[jerry@CentOS project_repo]$ ls
branches  tags  trunk

[jerry@CentOS project_repo]$ svn copy trunk branches/jerry_branch
A         branches/jerry_branch

[jerry@CentOS project_repo]$ svn status
A  +    branches/jerry_branch

[jerry@CentOS project_repo]$ svn commit -m "Jerry's private branch"
Adding         branches/jerry_branch
Adding         branches/jerry_branch/README

Committed revision 9.
[jerry@CentOS project_repo]$
```

Now Jerry is working in his private branch. He adds sort operation for the array. Jerry's modified code looks like this.

```
[jerry@CentOS project_repo]$ cd branches/jerry_branch/

[jerry@CentOS jerry_branch]$ cat array.c
```

The above command will produce the following result.

```
#include <stdio.h>
#define MAX 16

void bubble_sort(int *arr, int n)
{
   int i, j, temp, flag = 1;
   for (i = 1; i < n && flag == 1; ++i) {
      flag = 0;
      for (j = 0; j < n - i; ++j) {
         if (arr[j] > arr[j + 1]) {
            flag = 1;
            temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
         }
      }
   }
}

void accept_input(int *arr, int n)
{
   int i;

   for (i = 0; i < n; ++i)
   scanf("%d", &arr[i]);
}

void display(int *arr, int n)
```

```
{
   int i;

   for (i = 0; i < n; ++i)
   printf("|%d| ", arr[i]);

   printf("\n");
}

int main(void)
{
   int i, n, key, ret, arr[MAX];

   printf("Enter the total number of elements: ");
   scanf("%d", &n);

   /* Error handling for array overflow */
   if (n >MAX) {
      fprintf(stderr, "Number of elements must be less than %d\n", MAX);
      return 1;
   }

   printf("Enter the elements\n");
   accept_input(arr, n);

   printf("Array has following elements\n");
   display(arr, n);

   printf("Sorted data is\n");
   bubble_sort(arr, n);
   display(arr, n);

   return 0;
}
```

Jerry compiles and tests his code and is ready to commit his changes.

```
[jerry@CentOS jerry_branch]$ make array
cc     array.c   -o array

[jerry@CentOS jerry_branch]$ ./array
```

The above command will produce the following result.

```
Enter the total number of elements: 5
Enter the elements
10
-4
2
7
9
Array has following elements
|10| |-4| |2| |7| |9|
Sorted data is
|-4| |2| |7| |9| |10|

[jerry@CentOS jerry_branch]$ svn status
?       array
M       array.c

[jerry@CentOS jerry_branch]$ svn commit -m "Added sort operation"
Sending        jerry_branch/array.c
Transmitting file data .
Committed revision 10.
```

Meanwhile, over in the trunk, Tom decides to implement search operation. Tom adds code for search operation and his code looks like this.

```
[tom@CentOS trunk]$ svn diff
```

The above command will produce the following result.

```
Index: array.c
===================================================================
--- array.c   (revision 10)
+++ array.c   (working copy)
@@ -2,6 +2,27 @@

 #define MAX 16

+int bin_search(int *arr, int n, int key)
+{
+   int low, high, mid;
+
+   low   = 0;
+   high   = n - 1;
+   mid   = low + (high - low) / 2;
+
+   while (low <= high) {
+      if (arr[mid] == key)
+         return mid;
+      if (arr[mid] > key)
+         high = mid - 1;
+      else
+         low = mid + 1;
+      mid = low + (high - low) / 2;
+   }
+
+   return -1;
+}
+
 void accept_input(int *arr, int n)
 {
    int i;
@@ -22,7 +43,7 @@

 int main(void)
 {
-   int i, n, arr[MAX];
+   int i, n, ret, key, arr[MAX];

    printf("Enter the total number of elements: ");
    scanf("%d", &n);
@@ -39,5 +60,16 @@
    printf("Array has following elements\n");
    display(arr, n);

+   printf("Enter the element to be searched: ");
+   scanf("%d", &key);
+
+   ret = bin_search(arr, n, key);
+   if (ret < 0) {
+      fprintf(stderr, "%d element not present in array\n", key);
+      return 1;
+   }
+
+   printf("%d element found at location %d\n", key, ret + 1);
+
    return 0;
 }
```

After reviewing, he commits his changes.

```
[tom@CentOS trunk]$ svn status
?       array
```

```
M       array.c

[tom@CentOS trunk]$ svn commit -m "Added search operation"
Sending        trunk/array.c
Transmitting file data .
Committed revision 11.
```

But Tom is curious about what Jerry has been doing in his private branch.

```
[tom@CentOS trunk]$ cd ../branches/
[tom@CentOS branches]$ svn up
A    jerry_branch
A    jerry_branch/array.c
A    jerry_branch/README

[tom@CentOS branches]$ svn log
------------------------------------------------------------------------
r9 | jerry | 2013-08-27 21:56:51 +0530 (Tue, 27 Aug 2013) | 1 line

Added sort operation
------------------------------------------------------------------------
```

By viewing the Subversion's log message, Tom found that Jerry implemented 'sort' operation. Tom implemented search operation using binary search algorithm, it always expects data in sorted order. But what if the user provides data in an unsorted order? In that situation, binary search operation will fail. So he decides to take Jerry's code to sort data before search operation. So he asks Subversion to merge code from Jerry's branch into trunk.

```
[tom@CentOS trunk]$ pwd
/home/tom/project_repo/trunk

[tom@CentOS trunk]$ svn merge ../branches/jerry_branch/
--- Merging r9 through r11 into '.':
U    array.c
```

After merging, array.c will look like this.

```
[tom@CentOS trunk]$ cat array.c
```

The above command will produce the following result.

```c
#include <stdio.h>
#define MAX 16

void bubble_sort(int *arr, int n)
{
   int i, j, temp, flag = 1;

   for (i = 1; i < n && flag == 1; ++i) {
      flag = 0;
      for (j = 0; j < n - i; ++j) {
         if (arr[j] > arr[j + 1]) {
            flag       = 1;
            temp       = arr[j];
            arr[j]      = arr[j + 1];
            arr[j + 1] = temp;
         }
      }
   }
}

int bin_search(int *arr, int n, int key)
{
   int low, high, mid;

   low   = 0;
```

```
   high  = n - 1;
   mid   = low + (high - low) / 2;

   while (low <= high) {
      if (arr[mid] == key)
         return mid;
      if (arr[mid] > key)
         high = mid - 1;
      else
         low = mid + 1;
      mid = low + (high - low) / 2;
   }
   return -1;
}

void accept_input(int *arr, int n)
{
   int i;

   for (i = 0; i < n; ++i)
      scanf("%d", &arr[i]);
}

void display(int *arr, int n)
{
   int i;
   for (i = 0; i < n; ++i)
      printf("|%d| ", arr[i]);
   printf("\n");
}

int main(void)
{
   int i, n, ret, key, arr[MAX];

   printf("Enter the total number of elements: ");
   scanf("%d", &n);

   /* Error handling for array overflow */
   if (n > MAX) {
      fprintf(stderr, "Number of elements must be less than %d\n", MAX);
      return 1;
   }

   printf("Enter the elements\n");
   accept_input(arr, n);

   printf("Array has following elements\n");
   display(arr, n);

   printf("Sorted data is\n");
   bubble_sort(arr, n);
   display(arr, n);

   printf("Enter the element to be searched: ");
   scanf("%d", &key);

   ret = bin_search(arr, n, key);
   if (ret < 0) {
      fprintf(stderr, "%d element not present in array\n", key);
      return 1;
   }

   printf("%d element found at location %d\n", key, ret + 1);

   return 0;
}
```

After compilation and testing, Tom commits his changes to the repository.

```
[tom@CentOS trunk]$ make array
cc     array.c   -o array

[tom@CentOS trunk]$ ./array
Enter the total number of elements: 5
Enter the elements
10
-2
8
15
3
Array has following elements
|10| |-2| |8| |15| |3|
Sorted data is
|-2| |3| |8| |10| |15|
Enter the element to be searched: -2
-2 element found at location 1

[tom@CentOS trunk]$ svn commit -m "Merge changes from Jerry's code"
Sending        trunk
Sending        trunk/array.c
Transmitting file data .
Committed revision 12.

[tom@CentOS trunk]$
```