

## Task 1

Find out the abbreviations for the following commands: `list`, `checkout`, `commit`, and `update`. How did you get this information?

**\$ svn help**

Available subcommands:

- add
- auth
- blame (praise, annotate, ann)
- cat
- changelist (cl)
- checkout (co)
- cleanup
- commit (ci)
- copy (cp)
- delete (del, remove, rm)
- diff (di)
- export
- help (?, h)
- import
- info
- list (ls)
- lock
- log
- merge
- mergeinfo
- mkdir
- move (mv, rename, ren)
- patch
- propdel (pdel, pd)
- propedit (pedit, pe)
- propget (pget, pg)
- proplist (plist, pl)
- propset (pset, ps)
- relocate
- resolve
- resolved
- revert
- status (stat, st)
- switch (sw)
- unlock
- update (up)
- upgrade

## Task 2

Using the `-m` flag to provide a commit message on the command line is convenient, but it does not work well for giving longer, more detailed messages. How can you use a text editor to provide this message instead of using `-m`?

The way I did is create a txt named `changes.txt` under current directory, change the `doc/spec.txt` file and then write corresponding message in `changes.txt`

```
$ svn commit -F changes.txt
```

```
$ svn update
```

```
$ svn log Makefile
```

Then all the message related to changes that have been made will be displayed.

### Task 3

Run `svn status` from the `trunk` directory. What does “?” mean in the output? What do you think you should do in this case?

“?” means the Item is not under version control.

In this case, I should let Subversion know about the new file, by placing it under version control:

```
$ svn add changes.txt  
$ svn add bin/main
```

## Task 4

There will be a message saying that there is a conflict discovered in file 'src/fib.c'. I changed fib.c file by keeping my version. Use option 'r' to make the conflict marked as resolved. Then, output message becomes "Summary of conflicts: Text conflicts: 0 remaining (and 1 already resolved)." The last step is commit my-branch and the update my-branch.

Paste the code below:

```
[mingqi@royal-03] (92)$ svn merge -r 4:9
$REPO/project/branches/stack-branch
--- Merging r5 through r9 into '.':
U    include/stack.h
C    src/fib.c
U    src/stack.c
--- Recording mergeinfo for merge of r5 through r9 into '.':
U    .
Summary of conflicts:
  Text conflicts: 1
Conflict discovered in file 'src/fib.c'.
Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,
        (mc) my side of conflict, (tc) their side of conflict,
        (s) show all options: df
--- src/fib.c.working      - MINE
+++ src/fib.c              - MERGED
@@ -1,7 +1,17 @@
    #include <stdio.h>

    int fib(int n) {
+<<<<<<< .working

        return fib(n-2) + fib(n-1);

+||||||| .merge-left.r4
+    printf("fib: unimplemented!\n");
+    return -1;
+=====
+    if (n == 0) return 0;
+    if (n == 1) return 1;
+    printf("fib(%d): unimplemented!\n", n);
+    return -1;
+>>>>>>> .merge-right.r9
    }
```

Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,  
(r) mark resolved, (mc) my side of conflict,  
(tc) their side of conflict, (s) show all options: e

Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,  
(r) mark resolved, (mc) my side of conflict,  
(tc) their side of conflict, (s) show all options: s

(e) - change merged file in an editor [edit]  
(df) - show all changes made to merged file  
(r) - accept merged version of file [working]

(dc) - show all conflicts (ignoring merged version)  
(mc) - accept my version for all conflicts (same) [mine-conflict]  
(tc) - accept their version for all conflicts (same)

[theirs-conflict]

(mf) - accept my version of entire file (even non-conflicts)  
[mine-full]  
(tf) - accept their version of entire file (same) [theirs-full]

(m) - use merge tool to resolve conflict  
(l) - launch external merge tool to resolve conflict [launch]  
(i) - use built-in merge tool to resolve conflict  
(p) - mark the conflict to be resolved later [postpone]  
(q) - postpone all remaining conflicts  
(s) - show this list (also 'h', '?')

Words in square brackets are the corresponding --accept option arguments.

Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,  
(r) mark resolved, (mc) my side of conflict,  
(tc) their side of conflict, (s) show all options: e

Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,  
(r) mark resolved, (mc) my side of conflict,  
(tc) their side of conflict, (s) show all options: m

**Merging 'src/fib.c'.**

**Conflicting section found during merge:**

**(1) their version (at line 7) | (2) your version (at line 6)**

```
-----+-----  
-----  
    if (n == 0) return 0;          |
```

```

if (n == 1) return 1;                | return fib(n-2) + fib(n-1);

printf("fib(%d): unimplemented!\n", n|

return -1;                          |

```

```

-----+-----
-----

```

Select: (1) use their version, (2) use your version,  
 (12) their version first, then yours,  
 (21) your version first, then theirs,  
 (e1) edit their version and use the result,  
 (e2) edit your version and use the result,  
 (eb) edit both versions and use the result,  
 (p) postpone this conflicting section leaving conflict markers,  
 (a) abort file merge and return to main menu: 2

Merge of 'src/fib.c' completed.

Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,  
 (r) mark resolved, (mc) my side of conflict,  
 (tc) their side of conflict, (s) show all options: r

Resolved conflicted state of 'src/fib.c'

Summary of conflicts:

Text conflicts: 0 remaining (and 1 already resolved)

[mingqi@royal-03] (93)\$ svn commit -m "Complete merged"

Sending

.

Sending

include/stack.h

Sending

src/stack.c

Transmitting file data ..done

Committing transaction...

Committed revision 15.

## Task 5

```
[mingqi@royal-03] (106)$ svn blame src/stack.c
1      bob #include <stdio.h>
1      bob #include <stdlib.h>
1      bob
5      sally #define SIZE 50
5      sally
5      sally int stack[SIZE], *tos = stack, *p1 = stack;
5      sally
1      bob void push(int i) {
5      sally     p1++;
5      sally     if(p1 == (tos+SIZE)) {
5      sally         printf("Stack Overflow.\n");
5      sally         exit(1);
5      sally     }
5      sally     *p1 = i;
1      bob }
1      bob
1      bob int pop(void) {
7      sally     if(p1 == tos) {
7      sally         printf("Stack Underflow.\n");
7      sally         exit(1);
7      sally     }
7      sally     p1--;
7      sally     return *(p1+1);
1      bob }
1      bob
1      bob int peek(void) {
9      sally     return *p1;
1      bob }
1      bob
1      bob
```

## Task 6

In my-branch, I use '\$svn merge -r 9:8 \$REPO/project/branches/stack-branch' to undo the change made in version 9

Then, commit the change using '\$svn commit -m "undo revision 9."' and update.

```
svn update
```

```
Updating '.':
```

```
At revision 15.
```

```
[mingqi@royal-03] (124)$ svn merge -r 9:8 $REPO/project/branches/stack-branch
```

```
--- Reverse-merging r9 into '.':
```

```
U      src/stack.c
```

```
--- Recording mergeinfo for reverse merge of r9 into '.':
```

```
U .
```

```
[mingqi@royal-03] (125)$ ls
```

```
bin/ doc/ include/ Makefile src/
```

```
[mingqi@royal-03] (126)$ cd ..
```

```
[mingqi@royal-03] (127)$ ls
```

```
my-branch/ stack-branch/ toyrepo/ toyrepo.dumpfile trunk/
```

```
[mingqi@royal-03] (128)$ cd my-branch
```

```
[mingqi@royal-03] (129)$ svn blame src/stack.c
```

```
1      bob #include <stdio.h>
1      bob #include <stdlib.h>
1      bob
1      bob #define SIZE 50
1      bob
1      bob int stack[SIZE], *tos = stack, *p1 = stack;
1      bob
1      bob void push(int i) {
1      bob     p1++;
1      bob     if(p1 == (tos+SIZE)) {
1      bob         printf("Stack Overflow.\n");
1      bob         exit(1);
1      bob     }
1      bob     *p1 = i;
1      bob }
1      bob
1      bob int pop(void) {
1      bob     if(p1 == tos) {
1      bob         printf("Stack Underflow.\n");
1      bob         exit(1);
```



```
1      bob  }
1      bob  p1--;
1      bob  return *(p1+1);
1      bob }
15     mingqi
15     mingqi int peek(void) {
-      -   printf("peek: unimplemented\n");
-      -   return 0;
15     mingqi }
15     mingqi
```

15 mingqi

[mingqi@royal-03] (130)\$svn commit -m "undo revision 9"

[mingqi@royal-03] (130)\$svn update

## Task 7

```
$ svn checkout $REPO/project/branches/test-branch
$ svn merge $REPO/project/branches/my-branch
$ svn commit -m "merged stack and fib"
$ svn update
$ svn merge -r 8:9 $REPO/project/branches/stack-branch
$ svn commit -m "sally's implementation of peek"
$ svn update
$ cd ../trunk
$ svn merge $REPO/project/branches/test-branch
$ svn commit -m "test branch"
$ svn update
```

```
[mingqi@royal-03] (194)$ svn merge $REPO/project/branches/test-branch
```

```
--- Merging differences between repository URLs into '.':
```

```
U      include/stack.h
U      src/fib.c
U      src/main.c
U      src/stack.c
U      .
```

```
--- Recording mergeinfo for merge between repository URLs into '.':
```

```
G      .
```

```
[mingqi@royal-03] (195)$ svn commit -m "test branch"
```

```
Sending      .
Sending      include/stack.h
Sending      src/fib.c
Sending      src/main.c
Sending      src/stack.c
```

```
Transmitting file data ....done
```

```
Committing transaction...
```

```
svn Committed revision 20.
```

```
up[mingqi@royal-03] (196)$ svn update
```

```
Updating '.':
```

```
At revision 20.
```

```
[mingqi@royal-03] (197)$ svn diff -r 2:HEAD
```

```
Index: bin/main
```

```
=====
```

```
Cannot display: file marked as a binary type.
```

```
svn:mime-type = application/octet-stream
```

Index: bin/main

=====

--- bin/main (nonexistent)

+++ bin/main (revision 20)

Property changes on: bin/main

---

Added: svn:executable

## -0,0 +1 ##

+\*

\ No newline at end of property

Added: svn:mime-type

## -0,0 +1 ##

+application/octet-stream

\ No newline at end of property

Index: include/stack.h

=====

--- include/stack.h (revision 2)

+++ include/stack.h (revision 20)

@@ -1,3 +1,4 @@

+

/\* Global stack implementation \*/

void push(int k);

@@ -4,4 +5,5 @@

int pop();

+int peek();

Index: src/fib.c

=====

--- src/fib.c (revision 2)

+++ src/fib.c (revision 20)

@@ -1,6 +1,7 @@

#include <stdio.h>

int fib(int n) {

- printf("fib: unimplemented!\n");

- return -1;

+

```
+ return fib(n-2) + fib(n-1);
+
}
```

Index: src/main.c

```
=====
```

```
--- src/main.c (revision 2)
+++ src/main.c (revision 20)
@@ -12,8 +12,8 @@
```

```
        value = fib(choice);
```

```
-     if (choice > 1) push(value);
-     else printf("value on top is %d\n", pop());
+     if (choice > 1) push(value);
+     else if (choice >= 0) printf("value on top is %d\n", pop());
```

```
    } while(choice != -1);
```

Index: src/stack.c

```
=====
```

```
--- src/stack.c (revision 2)
+++ src/stack.c (revision 20)
@@ -1,21 +1,6 @@
#include <stdio.h>
#include <stdlib.h>
```

```
-void push(int i) {
- printf("push(%d): unimplemented\n", i);
-}
-
-int pop(void) {
- printf("pop: unimplemented\n");
- return 0;
-}
-
-int peek(void) {
- printf("peek: unimplemented\n");
- return 0;
-}
-
-/*
```

```
#define SIZE 50
```

```
int stack[SIZE], *tos = stack, *p1 = stack;
```

```
@@ -37,4 +22,9 @@
```

```
    p1--;
```

```
    return *(p1+1);
```

```
}
```

```
*/
```

```
+
```

```
+int peek(void) {
```

```
+    return *p1;
```

```
+}
```

```
+
```

```
+
```

```
Index: Makefile
```

```
=====
```

```
--- Makefile (nonexistent)
```

```
+++ Makefile (revision 20)
```

```
@@ -0,0 +1,17 @@
```

```
+CC=gcc
```

```
+CFLAGS=-Wall
```

```
+
```

```
+all: bin/main.o bin/stack.o bin/fib.o
```

```
+ $(CC) $(CFLAGS) -o bin/main bin/main.o bin/fib.o bin/stack.o
```

```
+
```

```
+bin/main.o: src/main.c include/stack.h include/fib.h
```

```
+ $(CC) $(CFLAGS) -Iinclude/ -c src/main.c -o bin/main.o
```

```
+
```

```
+bin/stack.o:
```

```
+ $(CC) $(CFLAGS) -c src/stack.c -o bin/stack.o
```

```
+
```

```
+bin/fib.o:
```

```
+ $(CC) $(CFLAGS) -c src/fib.c -o bin/fib.o
```

```
+
```

```
+clean:
```

```
+ rm -f bin/main bin/main.o bin/stack.o bin/fib.o
```

```
Index: .
```

```
=====
```

```
--- . (revision 2)
```

```
+++ . (revision 20)
```

Property changes on: .

---

Added: svn:mergeinfo

## -0,0 +0,3 ##

    Merged /project/branches/test-branch:r10-19

    Merged /project/branches/my-branch:r13-16

    Merged /project/branches/stack-branch:r5-9

[mingqi@royal-03] (198)\$

## Task 8

Unpack the two tar files and put them into a directory named "project" and do 'git branch'

fibonacci

\* master

stack

```
$ git checkout fibonacci          //switch to branch fibonacci
```

```
$ git diff master..HEAD          //show the difference between fib.c and main.c
```

```
$ git checkout stack            //switch to branch stack
```

```
$git diff master..HEAD //show the difference between stack.c and main.c
```

```
$git checkout master //switch to master stack
```

```
$git merge fibonacci
```

```
$git merge stack //there will be conflicts
```

```
$vim src/main.c //deal with the conflicts
```

```
$git add main.c
```

```
$make                          //run makefile)
```

```
$bin/main                      //run program
```

```
$git commit -m "final version"
```

```
$git merge stack              //already up to date
```

```
[mingqi@royal-03] (35)$ git diff HEAD^ HEAD
```

```
diff --git a/src/main.c b/src/main.c
```

```
index 7f2443c..020800c 100644
```

```
--- a/src/main.c
```

```
+++ b/src/main.c
```

```
@@ -4,17 +4,18 @@
```

```
#include "fib.h"
```

```
int main(void) {
```

```
-    int choice, value;
```

```
+
```

```
+    int choice;
```

```
    do {
```

```
        printf("Enter value: ");
```

```
        scanf("%d", &choice);
```

```

-     value = fib(choice);

-     if (choice > 1) push(value);
-     else if (choice >= 0) printf("value on top is %d\n", pop());
+     if (choice > 1) push(choice);
+     else printf("value on top is %d\n", pop());

    } while(choice != -1);
+
    return 0;
}
diff --git a/src/stack.c b/src/stack.c
index 8e3d58a..c18e06c 100644
--- a/src/stack.c
+++ b/src/stack.c
@@ -1,17 +1,28 @@
#include <stdio.h>
#include <stdlib.h>

+#define SIZE 50
+
+int stack[SIZE], *tos = stack, *p1 = stack;
+
void push(int i) {
-    printf("push(%d): unimplemented\n", i);
+    p1++;
+    if(p1 == (tos+SIZE)) {
+        printf("Stack Overflow.\n");
+        exit(1);
+    }
+    *p1 = i;
}

int pop(void) {
-    printf("pop: unimplemented\n");
-    return 0;
+    if(p1 == tos) {
+        printf("Stack Underflow.\n");
+        exit(1);
+    }
+    p1--;
+    return *(p1+1);
}

int peek(void) {
-    printf("peek: unimplemented\n");

```