



## Lab 6

# CS 5-1 - BSCS – Operating System Lab

**Name:** Sharaiz Ahmed

**SAP:** 57288

Task 1:

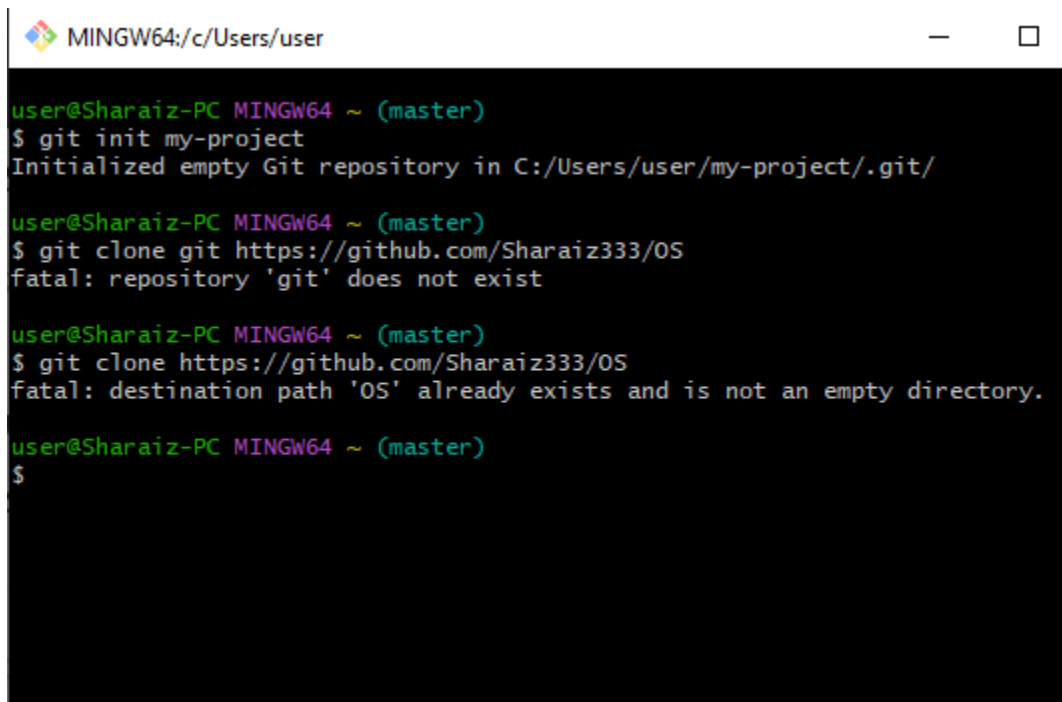
```
MINGW64:/c/Users/user
user@Sharaiz-PC MINGW64 ~ (master)
$ git config --global user.name "Sharaiz333"

user@Sharaiz-PC MINGW64 ~ (master)
$ git config --global user.email "57288@students.riphah.edu.pk"

user@Sharaiz-PC MINGW64 ~ (master)
$ git config --global color.ui auto

user@Sharaiz-PC MINGW64 ~ (master)
$
```

## Task 2:



A terminal window titled "MINGW64:/c/Users/user" with standard window controls. The terminal shows a series of Git commands and their outputs. The first command, `git init my-project`, succeeds. The second, `git clone git https://github.com/Sharaiz333/OS`, fails with the message "fatal: repository 'git' does not exist". The third, `git clone https://github.com/Sharaiz333/OS`, fails with "fatal: destination path 'OS' already exists and is not an empty directory." The prompt returns to the shell.


```
user@Sharaiz-PC MINGW64 ~ (master)
$ git init my-project
Initialized empty Git repository in C:/Users/user/my-project/.git/

user@Sharaiz-PC MINGW64 ~ (master)
$ git clone git https://github.com/Sharaiz333/OS
fatal: repository 'git' does not exist

user@Sharaiz-PC MINGW64 ~ (master)
$ git clone https://github.com/Sharaiz333/OS
fatal: destination path 'OS' already exists and is not an empty directory.

user@Sharaiz-PC MINGW64 ~ (master)
$
```

### Task 3:

 MINGW64:/c/Users/user

```
warning: could not open directory 'Templates/': Permission denied
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
.VirtualBox/
.android/
.bash_history
.config/
.emulator_console_auth_token
.gitconfig
.gitignore
.m2/
.nbi/
3D Objects/
AppData/
AppMods/
Contacts/
Desktop/
Downloads/
Favorites/
Intel/
IntelGraphicsProfiles/
Links/
Music/
NTUSER.DAT
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.b1f
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer000000000000
00000001.regtrans-ms
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer000000000000
00000002.regtrans-ms
OS/
OneDrive/
Oracle/
Recorded Calls/
Saved Games/
Searches/
StudioProjects/
Videos/
my-project/
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
MINGW64:/c/Users/user/os
user@Sharaiz-PC MINGW64 ~/os (main)
$ git init
Reinitialized existing Git repository in C:/Users/user/OS/.git/

user@Sharaiz-PC MINGW64 ~/os (main)
$ git add README.md
fatal: pathspec 'README.md' did not match any files

user@Sharaiz-PC MINGW64 ~/os (main)
$ git add .

user@Sharaiz-PC MINGW64 ~/os (main)
$ git diff

user@Sharaiz-PC MINGW64 ~/os (main)
$ git commit -m "Initial Commit"
On branch main

Initial commit

nothing to commit (create/copy files and use "git add" to track)

user@Sharaiz-PC MINGW64 ~/os (main)
$ |
```

#### Task 4:

```
MINGW64:/c/Users/user/os
You do not have the initial commit yet

user@Sharaiz-PC MINGW64 ~/os (main)
$ git commit
On branch main

Initial commit

nothing to commit (create/copy files and use "git add" to track)

user@Sharaiz-PC MINGW64 ~/os (main)
$ git stash
You do not have the initial commit yet

user@Sharaiz-PC MINGW64 ~/os (main)
$ git stash pop
No stash entries found.

user@Sharaiz-PC MINGW64 ~/os (main)
$ git stash drop
No stash entries found.

user@Sharaiz-PC MINGW64 ~/os (main)
$
```

## Task 5:

```
MINGW64:/c/Users/user/os
user@Sharaiz-PC MINGW64 ~/os (main)
$ git checkout -b hotfix
Switched to a new branch 'hotfix'

user@Sharaiz-PC MINGW64 ~/os (hotfix)
$ git merge branchname
merge: branchname - not something we can merge

user@Sharaiz-PC MINGW64 ~/os (hotfix)
$ git merge hotfix
merge: hotfix - not something we can merge

user@Sharaiz-PC MINGW64 ~/os (hotfix)
$ git rebase
There is no tracking information for the current branch.
Please specify which branch you want to rebase against.
See git-rebase(1) for details.

    git rebase '<branch>'

If you wish to set tracking information for this branch you can do so with:
```

## Task 6:

```
MINGW64:/c/Users/user/os
user@Sharaiz-PC MINGW64 ~ (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

user@Sharaiz-PC MINGW64 ~ (master)
$ cd os
,
user@Sharaiz-PC MINGW64 ~/os (hotfix)
$ git log ref..
fatal: ambiguous argument 'ref..': unknown revision or path not in the working t
ree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

user@Sharaiz-PC MINGW64 ~/os (hotfix)
$ git ref
git: 'ref' is not a git command. See 'git --help'.

The most similar commands are
    grep
    reflog

user@Sharaiz-PC MINGW64 ~/os (hotfix)
$ git reflog
```

## Task 7:

```
user@Sharaiz-PC MINGW64 ~ (master)
$ git tag

user@Sharaiz-PC MINGW64 ~ (master)
$ git tag v1.0

user@Sharaiz-PC MINGW64 ~ (master)
$ git log --online
fatal: unrecognized argument: --online

user@Sharaiz-PC MINGW64 ~ (master)
$ git log --online
fatal: unrecognized argument: --online

user@Sharaiz-PC MINGW64 ~ (master)
$ git tag -a v2.0 -m "Version 2.0 release"

user@Sharaiz-PC MINGW64 ~ (master)
$ git tag -d v1.0
Deleted tag 'v1.0' (was d057a08)

user@Sharaiz-PC MINGW64 ~ (master)
$ git tag
v2.0
```

## Task 8:

```
user@Sharaiz-PC MINGW64 ~ (master)
$ git reset [--hard] [target reference]

user@Sharaiz-PC MINGW64 ~ (master)
$ git revert [commit sha]
fatal: bad revision '[commit]'

user@Sharaiz-PC MINGW64 ~ (master)
$ git revert
usage: git revert [--[no-]edit] [-n] [-m <parent-number>] [-s] [-S[<keyid>]] <commit>...
or: git revert (--continue | --skip | --abort | --quit)

--quit                end revert or cherry-pick sequence
--continue            resume revert or cherry-pick sequence
--abort               cancel revert or cherry-pick sequence
--skip               skip current commit and continue
--[no-]cleanup <mode> how to strip spaces and #comments from message
-n, --no-commit       don't automatically commit
--commit             opposite of --no-commit
-e, --[no-]edit       edit the commit message
-s, --[no-]signoff    add a Signed-off-by trailer
-m, --[no-]mainline <parent-number>
                    select mainline parent
--[no-]rerere-autoupdate
                    update the index with reused conflict resolution if possible
--[no-]strategy <strategy>
                    merge strategy
-X, --[no-]strategy-option <option>
                    option for merge strategy
-S, --[no-]gpg-sign[=<key-id>]
                    GPG sign commit
--[no-]reference      use the 'reference' format to refer to commits

user@Sharaiz-PC MINGW64 ~ (master)
$ git revert v2.0
```

## Task 9:

```
user@Sharaiz-PC MINGW64 ~ (master)
$ git fetch

user@Sharaiz-PC MINGW64 ~ (master)
$ git pull
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=<remote>/<branch> master

user@Sharaiz-PC MINGW64 ~ (master)
$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository us
ing

    git remote add <name> <url>

and then push using the remote name

    git push <name>

user@Sharaiz-PC MINGW64 ~ (master)
$ git push v2.0
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream v2.0 master

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

## Task 10:

## Task 11:

```
user@Sharaiz-PC MINGW64 ~ (master)
$ cat - <<EOF > .gitignore
> /logs-*
!logs/.gitkeep
/tmp
*.swp
EOF

user@Sharaiz-PC MINGW64 ~ (master)
$ |
```