2810ICT Software Technologies

Workshop 4 – Software Configuration Management

Objectives:

In this workshop you will:

- Setup a version control system using GIT and BitBucket
- Make some modifications to a program and commit them to the local repository
- Push the local rep to the BitBucket account
- Revert back to a previous version of your software

Overview

The lecture in this week covers software configuration management. For this workshop, we will be using git from the command line. You will be required to take some python code, commit it to the local repository and push it to BitBucket as a baseline. You will then modify your code and push a new version to the server. Finally you will need to retrieve an old version to undo the latest changes.

Task 1 – GIT (3 marks)

Step 1 – Create a BitBucket account and repository

- Go to www.bitbucket.org and sign up for a free account. We recommend you use your student email address, but you may use another one if you prefer.
- Now create a new repository (call it workshop5 or similar). The default settings (private, GIT) are fine. For now you do not need to add any files to the repo.

Step 2 – Create a local repository

- In a command prompt, navigate to a directory (CD) where you would like to create your local repository. This can be temporary (desktop) or on your Z drive.
- Clone the remote repository to your local machine

```
git clone https://username@bitbucket.org/username/reponame.git
```

- You should replace "username" with your bitbucket username and "reponame" with whatever you called your repository. You will automatically be asked for your password.

Step 3 – Add the initial python files to the local repository, then push it to the server

- Copy countWords.py and histogram.py (written in Workshop 3) into the repository directory created locally (this will be a folder called 'reponame' in whatever location you executed your git clone command.)
- Change to your repo directory (cd .\reponame) and add the two files to your local repo

```
git add .\countWords.py
```

- Commit this change to your local repository

```
git commit –m "Initial Python Programs added"
```

- Push your local repository to the remote server

```
git push -u origin master
```

Step 4 – Modify your local programs

- You should now modify your local countWords.py program to accept mbox-short.txt as input. This file can be downloaded from the PyLab page on Learning@Griffith.
- Commit the changes (remember to use -m to leave a comment describing what you changed) and then push the changes to the master.

Step 5 – Revert back to the iterative method

- Typically when making changes that are experimental, we would start new branches. In this case, we have performed everything on one branch, but we can still retrieve our original files. View the log of what has happened to the repo by typing "git log"
- You should see an output that shows the 2 commits and (importantly) their hashes (ie: commit fdaf794b1fb5cfc8efc4017200f75d9414cc2503)
- To go back to a previous version, we want to use the "revert" command to undo an update.
- Find the hash that corresponds to the update you made (the most recent one), then use the command:

```
git revert --no-edit <hash>
```

- This will undo the changes introduced in that commit and revert your local repository to the previous state. You can now do a push to update the remote repository.

Task 2 – Shell Scripting (2 mark)

For each user, there is a folder called public_htmls in the home directory of dwarf server (if you have accidently deleted it, you can create one by yourself). This folder is used to hold the homepage of a user. Once the homepage is constructed, it can be accessed via the following link: http://dwarf.ict.griffith.edu.au/~sxxxxxxx/

In order to get the homepage setup, you need to create an html page called index.html and put it in the public_htmls folder. Create one if you know how to do it. Otherwise, create a file with the following html code:

```
<html>
<body>
This is my first 2810ICT homepage.
</body>
</html>
```

Next, create a folder under public_htmls called "code". Copy your countWords.py and histogram.py into this code folder, so that other people can access them via http://dwarf.ict.griffith.edu.au/~sxxxxxxx/code.

Remember to properly set the access right of your folders.

Task 3 – Scripting (1 mark)

Extend the bash script and python programs in Task 4 of Workshop 3 by allowing two text files as the command line arguments. Complete the word count and histogram generation on these two text files jointly, i.e., only one histogram on all words from two files.