

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Інститут Прикладного системного аналізу
Кафедра Системного проектування

Лабораторна робота №1
з дисципліни “Проектування інформаційних
систем”
«Системи контролю версій»

Виконав:
студент групи
ДА-71
Факультету «ІПСА»
Бідалов Володимир

Мета роботи: за допомогою системи контролю версій завантажити коди програми у репозиторій. Відтворити типовий цикл розробки програмного забезпечення з використанням системи контролю версій.

1. Обрати безкоштовну систему репозиторія для системи контролю версіями, наприклад projectlocker, або інш.

Для виконання лабораторних робіт була обрана система контролю версій git та безкоштовний онлайн-репозиторій GitHub.

2. Встановити клієнтське безкоштовне програмне забезпечення для роботи з системою контролю версій (GIT, SVN clients).

Було встановлено консольний клієнт git.

Його основні команди:

git clone - створює локальну копію віддаленого репозиторію.

git status - відображення поточного стану репозиторію, які файли були змінені та які відслідковуються чи ні.

git checkout - перенесення вказівника HEAD (куди дивиться робоча директорія), параметр -u вказує, що потрібно створити нову гілку.

git add - додає всі поточні зміни в індекс (увійдуть в наступний коміт).

git commit - зберігає всі індексовані зміни у новостворений коміт, переміщає поточний вказівник на цей коміт.

git push - публікує локальні зміни у віддалений репозиторій.

Створимо власну папку з файлом-прикладом в окремій гілці у репозиторії групи за допомогою цих команд:

```
just@rgarch ~/kpi/isd$ git clone git@github.com:TarasVolovodenko/da71.git
Cloning into 'da71'...
remote: Enumerating objects: 108, done.
remote: Counting objects: 100% (108/108), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 313 (delta 26), reused 62 (delta 10), pack-reused 205
Receiving objects: 100% (313/313), 5.49 MiB | 3.20 MiB/s, done.
Resolving deltas: 100% (85/85), done.
just@rgarch ~/kpi/isd$ cd da71
just@rgarch ~/kpi/isd/da71$ git checkout -b volodymyr_bidalov
Switched to a new branch 'volodymyr_bidalov'
just@rgarch ~/kpi/isd/da71$ volodymyr_bidalov take Bidalov
just@rgarch ~/kpi/isd/da71/Bidalov$ volodymyr_bidalov echo '## Volodymyr Bidalov' > README.md
just@rgarch ~/kpi/isd/da71/Bidalov$ volodymyr_bidalov git status
On branch volodymyr_bidalov
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  ./

nothing added to commit but untracked files present (use "git add" to track)
just@rgarch ~/kpi/isd/da71/Bidalov$ volodymyr_bidalov ls
README.md
just@rgarch ~/kpi/isd/da71/Bidalov$ volodymyr_bidalov git add .
just@rgarch ~/kpi/isd/da71/Bidalov$ volodymyr_bidalov git status
On branch volodymyr_bidalov
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   README.md

just@rgarch ~/kpi/isd/da71/Bidalov$ volodymyr_bidalov git commit -m '[vbidalov] README file'
[volodymyr_bidalov d2f8990] [vbidalov] README file
1 file changed, 1 insertion(+)
create mode 100644 Bidalov/README.md

just@rgarch ~/kpi/isd/da71/Bidalov$ volodymyr_bidalov git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 373 bytes | 373.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:TarasVolovodenko/da71.git
  55e7e47..cfb60cc volodymyr_bidalov -> volodymyr_bidalov
```

3. Описати цикл розробки програмного забезпечення з використанням системи контролю версій.

Процес роботи виглядає таким чином:

- Створення локальної копії проекту
- Зміна локальної копії
- Оновлення локальної копії при необхідності
- Фіксування локальних змін у вигляді комітів
- Публікація змін до віддаленого репозиторію

Висновок:

У ході лабораторної роботи було розглянуто роботу з системою контролю версій git. Вона є досить простою, її консольний клієнт має інтуїтивно зрозумілий інтерфейс, водночас ця система також є дуже гнучкою і легко адаптується під потреби як малих, так і великих команд розробників.