



CEID @ UNIVERSITY of PATRAS
Computer Engineering & Informatics Department

Εργαστήριο Μαθηματικών Θεμελιώσεων της Επιστήμης των
Υπολογιστών

Οδηγίες χρήσης του Github

Δημιουργία λογαριασμού

<https://github.com/join>

Λήψη και Εγκατάσταση του git

Εγκατάσταση σε Linux

Για διανομές βασισμένες σε Debian(όπως το ubuntu)

```
sudo apt-get install git-core
```

ενώ, για διανομές βασισμένες σε Fedora

```
yum install git-core
```

TIP: Με την εντολή `sudo aptitude search git` εμφανίζονται πακέτα που σχετίζονται με το git.

Εγκατάσταση σε Windows

Η εγκατάσταση στα windows είναι πολύ εύκολη! Κατεβάστε από τον σύνδεσμο <https://git-scm.com/download/win> το εκτελέσιμο αρχείο και τρέξτε το. Μόλις γίνει η εγκατάσταση θα πρέπει να έχετε ένα πρόγραμμα Git Bash, σαν ένα απλό κέλυφος όπου μπορείτε να εκτελείτε εντολές σχετικά με το git.

Για windows υπάρχει ακόμα ένα γραφικό περιβάλλον χρήσης. Αναλυτικές οδηγίες χρήσης του μπορείτε να βρείτε εδώ <https://ellak.gr/wiki/index.php?title=GitHub>

Εγκατάσταση σε Mac OS:

<https://git-scm.com/download/mac>

Τι είναι το git;

Το git είναι ένα σύστημα διαχείρισης εκδόσεων αρχείων:

- Στα Αγγλικά: version control system,
- Παρακολουθεί τις αλλαγές σε ένα σύνολο αρχείων κρατώντας στιγμιότυπά τους,
- Το σύνολο στιγμιότυπων ονομάζεται αποθετήριο (repository)

Στόχοι του Git είναι να υποστηρίξει τη διανεμημένη παραγωγικότητα και την ταχύτητα. Το Git βοηθά τον χρήστη να διατηρήσει την υπευθυνότητα και την πατρότητα για κάθε δράση του, να διατηρεί την ακεραιότητα του έργου και να βελτιώνει τη συνολική επικοινωνία μεταξύ των συνεργατών

Η σχεδιαστική φιλοσοφία αντικατοπτρίζεται στην κοινή φιλοσοφία χρήσης:

- Commit νωρίς και συχνά,
- κάθε commit αντιπροσωπεύει μία ιδέα ή μια ολόκληρη αλλαγή (και έτσι είναι ευκολότερη η ανάγνωση και η επαναφορά σε προηγούμενη έκδοση)
- Κάθε branch αντιπροσωπεύει ένα χαρακτηριστικό (ή θέμα) (και έτσι είναι ευκολότερη η ανάγνωση και η συγχώνευση) την τοπική σας

κατάλογο εργασίας, και ο δείκτης των τοπικών συμφωνιών επαναφοράς είναι τακάκια το μηδέν.

Χρήσεις του git:

- συμμετοχή σε κοινά project, και έργα ανοικτού λογισμικού
- επιστροφή σε μια παλιότερη έκδοση αν έχει γίνει κάποιο λάθος
- δοκιμή παραλλαγών (ονομάζονται παρακλάδια -- branches) χωρίς να επηρεάζεται η τρέχουσα κατάσταση των αρχείων

Τι είναι το GitHub

Το GitHub είναι ένας εξυπηρετητής που φιλοξενεί αποθετήρια και μια εύχρηστη διεπαφή ιστού

- υποστηρίζει όλες τις εντολές του git
- και παρέχει μερικές επιπλέον δυνατότητες

Είναι εξαιρετικά δημοφιλές. Επίσης παρέχει έλεγχο πρόσβασης και πολλές δυνατότητες συνεργασίας, όπως ανίχνευση σφαλμάτων, αιτήματα χαρακτηριστικών, διαχείριση εργασιών, και wikis για κάθε project.

Οι πρώτες Ρυθμίσεις

Στην εφαρμογή του git που εγκαταστήσατε στον υπολογιστή σας θα πρέπει να κάνετε κάποιες ρυθμίσεις, δηλώνοντας το όνομα σας και το email σας.

Πληκτρολογήστε (ή αντιγράψτε τις παρακάτω εντολές – αντικαθιστώντας τα Your Name και τα email@provider.com

```
git config --global user.name "Your Name"
git config --global user.email "email@provide.com"
```

Για email χρησιμοποιείτε τη διεύθυνση που δώσατε στο GitHub

Οδηγίες χρήσης του github

Δημιουργία τοπικού αποθετηρίου στον υπολογιστή σας

Για να αντιγράψουμε το project απο το github στον δίσκο μας εκτελούμε:

```
git clone https://github.com/HackTheBook/myhackthebookproject
```

*αντικαταστήστε το myhackthebookproject με το όνομα του repository της ομάδας σας.

Έτσι λοιπόν θα δημιουργηθεί ένας κατάλογος myhackthebookproject στον σκληρό σας.

Ο κατάλογος αυτός είναι ο κατάλογος στον οποίο θα εργάζεστε και θα βάζετε τα αρχεία για κοινή χρήση στο github.

Βασικές εντολές που πρέπει να γνωρίζετε

1. git-add

Προσθέτει το περιεχόμενο του αρχείου και το ετοιμάζει για να γίνει το επόμενο commit. Είναι και η πρώτη εντολή που θα πληκτρολογήσετε:

Χρήση εντολής

```
git add <filenames>  
πχ git add readme.txt
```

2. git commit

Η εντολή git commit ξεκινά έναν editor ώστε να γράψετε ένα σύντομο κείμενο που να εξηγεί τι περιλαμβάνει το commit:

- εναλλακτικά πληκτρολογήστε:

```
git commit --m "το μήνυμά σας (p.x add initial plan for myhackthebookproject)"
```

Τα κείμενα αυτά είναι σημαντικά

- πρέπει να μπορεί κανείς να καταλαβαίνει τι κάνατε (και γιατί)
- φανταστείτε ότι το γράφετε για κάποιον άλλο

3. git push

Με αυτή την εντολή ανεβάζετε οριστικά στο github τα αρχεία τα οποία κάνατε add και commit.

4. git-status

Περιγραφή: Εμφανίζει την κατάσταση του καταλόγου σας (ποια αρχεία είναι ήδη στο Github ποια αρχεία έχουν αλλάξει και ποια έχουν είναι εντελώς καινούργια και πρέπει να γίνουν πρώτα πρώτα add.

5. **git-diff**

Περιγραφή: Εμφανίζει διαφορές στα commit που έχουν γίνει και στο github

Περισσότερες πληροφορίες

Αναλυτική βοήθεια για την χρήση του github <https://help.github.com/>

Προχωρημένες εντολές

Exploring the repository

- `git log`

Εμφανίζει πληροφορίες για τα commit που έχουν γίνει.

- `git log --grep=<pattern> ή git log -S'<pattern>'`

Θα φιλτράρει τα commits από το συγκεκριμένο pattern.

- `git diff`

Εμφανίζει διαφορές στα commit που έχουν γίνει και στο working tree με τα index

- `git diff [<path>]`

αλλαγές σε σχέση με το INDEX

- `git diff <commit> [<commit>] [<path>]`

Αλλαγές μεταξύ δύο commit. Εάν παραλείπεται, το δεύτερο εκτελεί την προεπιλογή HEAD.

- `git show <commit>`

Θα εμφανίσει όλες τις αλλαγές που επήλθαν στο δεδομένο commit.

Εργασία με branches

- `git branch`

Εμφανίζει σε λίστα τα branches που υπάρχουν.

- `git branch <branch_name>`

Δημιουργεί νέο branch αλλά δεν αλλάζει σε αυτό.

- `git checkout <branch_name>`

Εναλλαγή μεταξύ branches.

- `git branch -d [head]`

Διαγραφή κάποιου branch όταν πάψει να μας χρειάζεται.

Merging: Αφού αναπτύξουμε τα νέα features στα ξεχωριστά branches πρέπει κάπως να τα εισάγουμε στο master branch:

```
git merge [head]
```

Τι κάνει το merge;

Εντοπίζει το κοινό πρόγονο (ancestor) του HEAD και του head που θα γίνει merge.

Αν ο πρόγονος == merge τότε δε κάνει τίποτα

Αν ο πρόγονος == HEAD τότε fast forward merge

Διαφορετικά εντοπίζει διαφορές μεταξύ πρόγονου και merge. Δοκιμάζει να τα συγχωνεύσει σε ένα αρχείο. Αν δεν υπάρχει σύγκρουση δημιουργεί νέο commit. Αν υπάρχει σύγκρουση δείχνει που υπάρχει πρόβλημα και ενημερώνει το χρήστη.

Merging Branches

```
git merge <commit>
```

Ενώνει δύο ή περισσότερα branches μαζί. Για παράδειγμα το master με κάποιο δεύτερο που δημιουργήσατε.

```
git merge --abort
```

Αναιρεί μια συγχώνευση σε εξέλιξη (πάντα κάνετε commit πριν από τη συγχώνευση).

```
git cherry-pick <commit>
```

Επιλέξτε μόνο μερικά commits για τη συγχώνευση.

Solving Conflicts

Όταν τα branches έχουν αλλάξει το ίδιο κομμάτι του κώδικα, προκύπτουν συγκρούσεις οι οποίες επιλύονται:

Χειροκίνητα ή χρησιμοποιώντας το εργαλείο mergetool:

```
git mergetool
```

Προσθέτοντας τα αρχεία στο INDEX. Κάνοντας commit στα αρχεία τα οποία έχετε επιλύσει τις συγκρούσεις.

Χρήσιμες πηγές

<http://tom.preston-werner.com/2011/03/29/ten-lessons-from-githubs-first-year.html>

<http://johnmclaughlin.info/the-philosophy-of-git-and-github/>