

2 LABORATORINIS DARBAS

Atliko: Lukas Serapinas, EKSfm-16

Sukursime git tipo projektų valdymo sistemos repozitorijas, kurios vienoje mašinoje (viename kompiuteryje) imituos 2 skirtingus vartotojus ir vieną serverį. Vartotojai keis projekto turinį (tiksliau jo failus) o pakeitimus galima saugoti serveryje.

Sukuriame direktorijas „home“ aplanke:

1. `mkdir server_side`
2. `mkdir client_side`

Įeiname į šių direktorių vidų ir kiekvienoje iš jų sukuriame „gitrepo“ direktoriją. Ši direktorija bus naudojama projektų valdymui ir bus inicializuota su git. Tuo tarpu nuotolinė jos kopija bus „server_side/gitrepo“ direktorijoje.

1. `cd server_side`
2. `mkdir gitrepo`
3. `cd ~/client_side`
4. `mkdir gitrepo`

Pereiname į „server_side“ direktoriją ir inicializuojame git.

1. `git init –bare`

Grįžtame į „client_side“ direktoriją (lokaliją), čia taip pat inicializuojame git, bei sukuriame tekstinį failą, o jame įrašome eilutę „Text file created by the first user.“

2. `cd ~/client_side`
3. `git init`

nano text.txt (čia įrašome eilutę, spaudžiame ctrl+X kombinaciją, pasirenkame „Y“ – kaip sutikimo ženklą ir spaudžiame ENTER). Pridedame nuotolinę repozitoriją su komanda:

1. `git remote add server_side ~/server_side/gitrepo/`

patvirtiname pakeitimus lokaliajoje repozitorijoje komandomis:

2. `git add text.txt`
3. `git commit –m ‘added text.txt file’`

ir išsiunčiame šios repozitorijos (tiksliau pagrindinės jos šakos) kopiją į nuotolinę repozitoriją „server_side“

1. `git push server_side master`

sukuriame naują vartotojo direktoriją (lokaliją) „client2_side“ ir inicializuojame ją su git:

2. `cd ~/`
3. `mkdir client2_side`
4. `cd client2_side && mkdir gitrepo && cd gitrepo`
5. `git init`

nurodome nuotolinės repozitorijos vietą (surišame šias repozitorijas), bei atsisiunčiame visą projektą (bei visus kada nors darytus pakeitimus)

1. `git add remote server_side ~/server_side/gitrepo/`
2. `git pull server_side master`

atsiradus `text.txt` bylai, joje įrašome naują eilutę su `nano` tvarkykle. Išsaugojame pakeitimus ir juos siunčiame į nuotolinę repozitoriją (tiksliau į pagrindinę jos šaką „`master`“):

1. `git add text.txt`
2. `git commit -m ,text file edited (added new line)'`
3. `git push server_side master`

Dabar imituosime konfliktą, kai vienas vartotojas pakeičia projektą vienaip, o kitas vartotojas – kitaip. Jeigu abu vartotojai bando pakeitimus saugoti serveryje, `git` įvyksta konfliktas. T.y. pakeitimai viename projekte įvyko nenuoseklia eiga, ir `git` nežino kokią paskutinę projekto būseną palikti.

`Client2_side` pridedame `text.txt` byloje eilutę „another line added by the second user“, o `Client_side` pridedame – „line added by the first user“. Šiuos pakeitimus išsaugojame su „`git add`“ bei „`git commit`“. Dabar siunčiame vieno iš šių vartotojų pakeitimus į serverį su „`git push server_side master`“. Šie pakeitimai atsiras pirmieji. Vėliau siunčiame antrojo vartotojo pakeitimus į serverį. Šį kartą atsitinka konfliktas ir siųsti nebeleidžiama. Todėl pirma padarome „`git pull server_side master`“, atsisunčiame pakeitimus išsaugotus serveryje. `Git` leidžia lengvai išspręsti atsiradusius konfliktus, pažymėdama kur jie tiksliai yra. Šiuo atveju pažymima paskutinė mūsų žinutė. Paredagavus ir sudėjus jas vieną po kitos, išsaugome pakeitimus ir siunčiame į server su „`git add`“, „`git commit`“ bei „`git push server_side master`“.