Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrarea de laborator#1

Version Control Systems si modul de setare a unui server

Autor:
Maria Mitriuc

lector asistent:
Irina Cojanu
lector superior:
Radu Melnic

Lucrare de laborator Nr.1

1 Scopul lucrarii de laborator

Scopul acestei lucrari de labolator este studierea Version Control Systems i modului de setare a unui server.

2 Obiective

Obiectivul principal al acestei lucrri de labolator este studierea si aplicarea n practic Version Control Systems, folosind una din platformele git, bitbucket, mercurial sau svn.

3 Implimentarea lucrarii de laborator

3.1 Sarcini si Obiective

1. Nivel
   – **initializeaza un nou repositoriu**
   La prima etapa cream un nou repozitoriu , urmind pasiii din ghidul de utilizare de pe platforma **github**, accesind butonul " + " si optiunea create new repository.
   – **configureaza-ti VCS**

   Pentru aceasta ar trebui sa introducem in linia de comanda gitbash urmatoarele caractere:
   ```
   git config --global user.name "YourName"
   git config --global user.email "youremail@domain.com
   ```

   Putem adauga fisiere in repozitoriu cu ajutorul comenzii: `git add` si sa verificam starea acestuia prin: `git status`
   – **crearea branch-urilor (creeaza cel putin 2 branches)**
   Pentru a crea o ramura in linia de comanda se executa urmatoarele comenzi:
   `git checkout -b branch1`-cream ramura
   `git branch -a`-vedem toate ramurile existente
   Introducem schimbari in ramura respectiva si – commit pe ambele

branch-uri (cel putin 1 commit per branch)

3.2 Taskuri si punctaj

Nivelul1 (nota 5||6) :
initializeaza un nou repositoriu
configureaza-ti VCS
ocrearea branch-urilor (creeaza cel putin 2 branches)
commit pe ambele branch-uri (cel putin 1 commit per branch)

Normal Level (nota 7||8):
seteaza un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)
reseteaza un branch la commit-ul anterior
salvarea temporara a schimbarilor care nu se vor face commit imediat.
folosirea fisierului .gitignore

Advanced Level (nota 9||10):
merge 2 branches
rezolvarea conflictelor a 2 branches
comezile git care trebuie cunoscute
Bonus Point (+1):
Tags. Folosirea tag-urilor pentru marcarea schimbarilor simnificative precum release-ul.

3.3 Analiza lucrarii de labolator

git@github.com:wedwer666/MIDPS.git - link la repositoriul pe github

1. Initializeaza un nou repositoriu am creat un repositoriu pe platforma github folosind email si parola.

2. Configureaza-ti VCS
2.1 crearea branch-urilor (creeaza cel putin 2 branches) realizarea acestui task prin comenzile git checkout b midps1 branche-ul numarul 1 si midps2 branche-ul numarul 2.
2.2 commit pe ambele branch-uri (cel putin 1 commit per branch) realizarea commitului a fost realizata prin crearea a 2 mapi cu ajutorul instructiunii

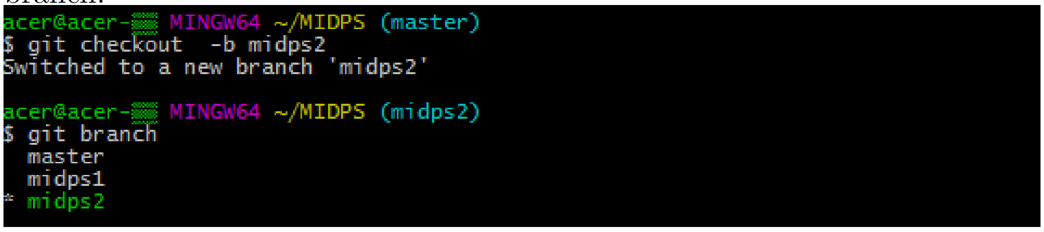
touch si inserarea codului prin vim sau vi.

3.1 Seteaza un branch to track a remote origin pe care vei putea sa faci push -aceasta a fost facut utiliind intructiunea git push origin "denumirea file-ului"
3.1 reseteaza un branch la commit-ul anterior - git reset "denumirea branchului dorit" sau folosind instructinunea git reset –soft "denumirea" sau git reset –hard in caz daca este nenesar.
3.2 salvarea temporara a schimbarilor care nu se vor face commit imediat - aceasta actiune se realizeaza folosind instructuinuea stash-git status, apio git checkout -b "denumirea" si ultima actiune git stash apply , dar fara a face commit la momentul de fata
3.3 folosirea fisierului .gitignore - acest fisier ignoreaza informatia care nu va fi dorit sa fie in commit afsisata
3.4 merge 2 branches - aceasta actiune a fost realizata folosind comanda merge "denumire 1 branch" "denumirea 2-lea branch"
3.5 rezolvarea conflictelor a 2 branches - git nu are ustensile pentru a rezolva acest conflict, din aceasta cauza conflictul aparut a fost rezolvat manual(in cazul dat nu a fost posibila rezolvarea automata)
3.6 comezile git care trebuie cunoscute - git config, git init, git clone, git add, git rm, git commit, git status, git branch, git checkout, git merge, git reset, git stash, git tag, git pull, git push , git remote, git log, git show, git cat-file, git grep,git diff, gitk, git archive, git gc, git fsck

4 Tags. Folosirea tag-urilor pentru marcarea schimbarilor simnificative precum release-ul

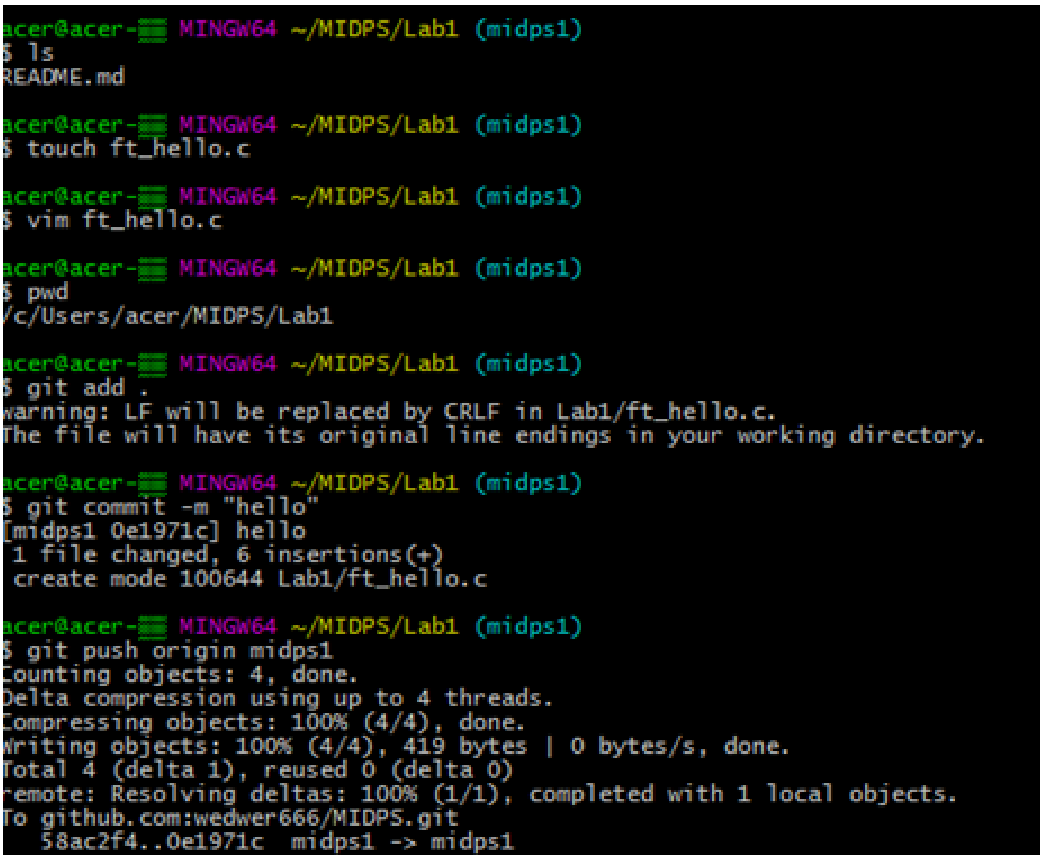
3.4 Imagini

Crearea a 2 branch-uri noi si vizualizarea acestora folosind comanda git branch:



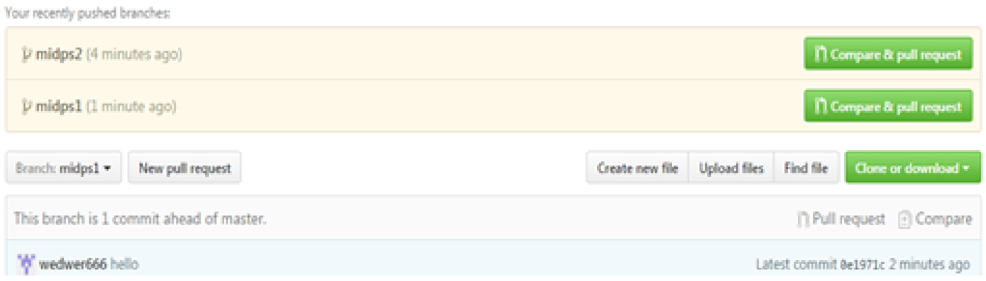
```
acer@acer-▒ MINGW64 ~/MIDPS (master)
$ git checkout  -b midps2
Switched to a new branch 'midps2'

acer@acer-▒ MINGW64 ~/MIDPS (midps2)
$ git branch
  master
  midps1
* midps2
```

Commit pe ambele branch-uri:



```
acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ ls
README.md

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ touch ft_hello.c

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ vim ft_hello.c

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ pwd
/c/Users/acer/MIDPS/Lab1

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ git add .
warning: LF will be replaced by CRLF in Lab1/ft_hello.c.
The file will have its original line endings in your working directory.

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ git commit -m "hello"
[midps1 0e1971c] hello
 1 file changed, 6 insertions(+)
 create mode 100644 Lab1/ft_hello.c

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ git push origin midps1
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 419 bytes | 0 bytes/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
To github.com:wedwer666/MIDPS.git
   58ac2f4..0e1971c  midps1 -> midps1
```

Rezultatele pe git:



Your recently pushed branches:

midps2 (4 minutes ago)

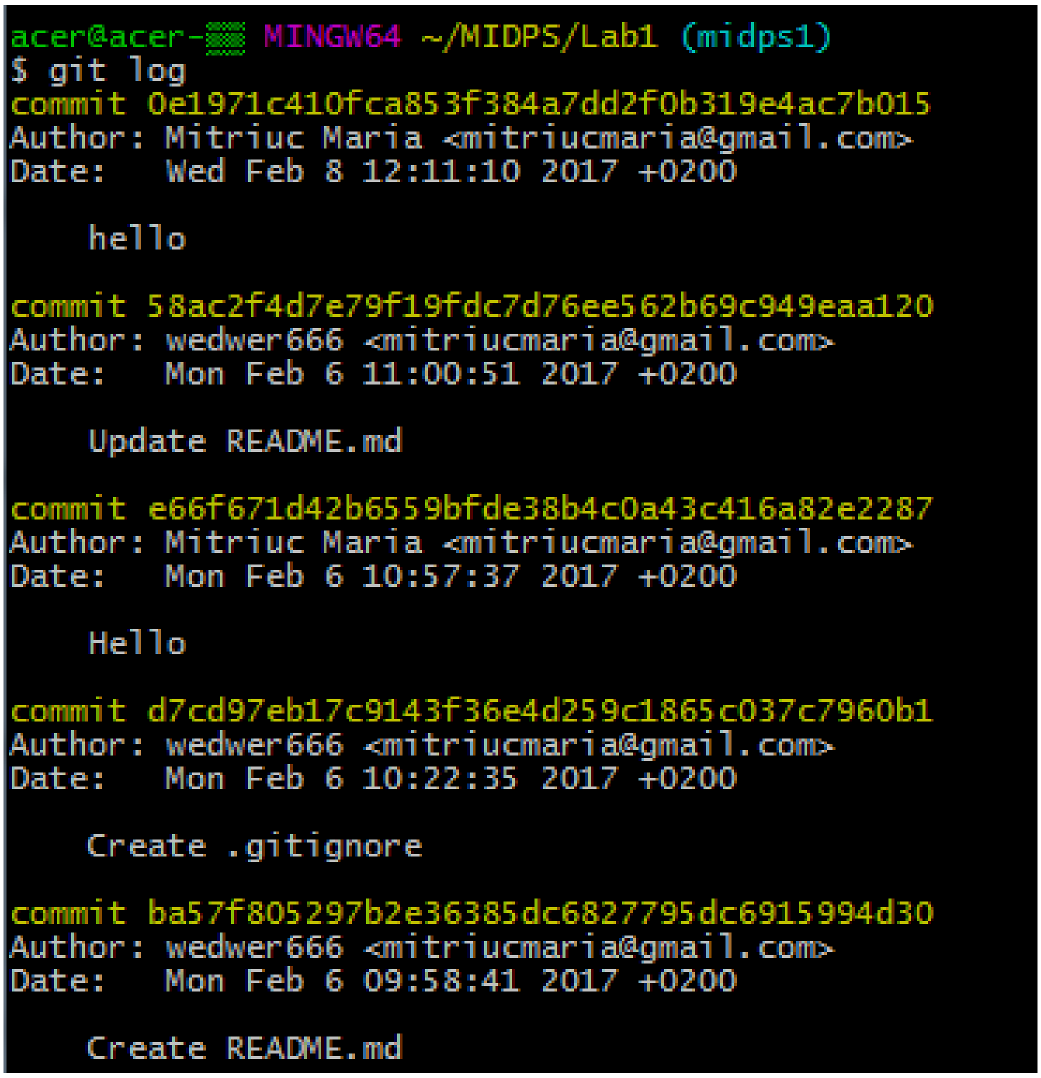
midps1 (1 minute ago)

This branch is 1 commit ahead of master.

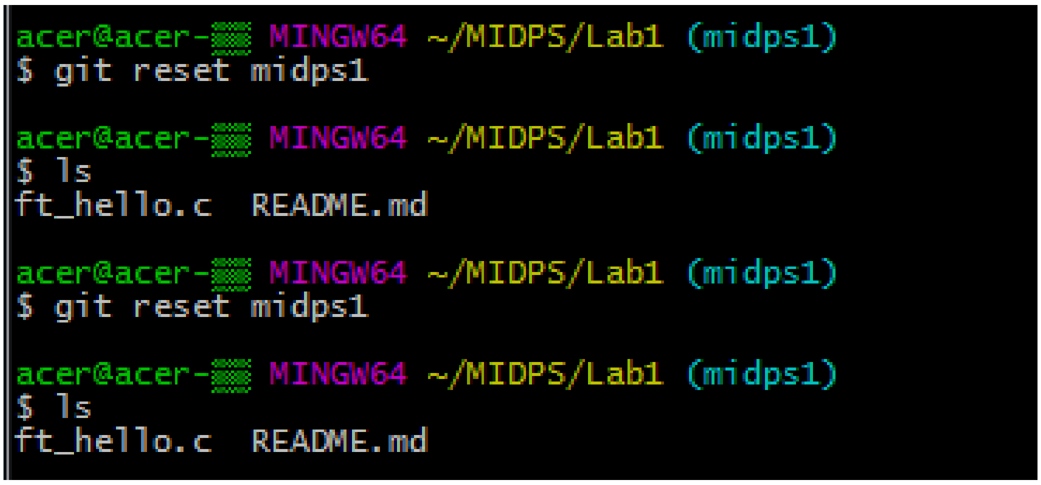
wedwer666 hello

Latest commit 0e1971c 2 minutes ago

Resetarea unui branch la commit-ul anterior: Folosirea comenzii git log:

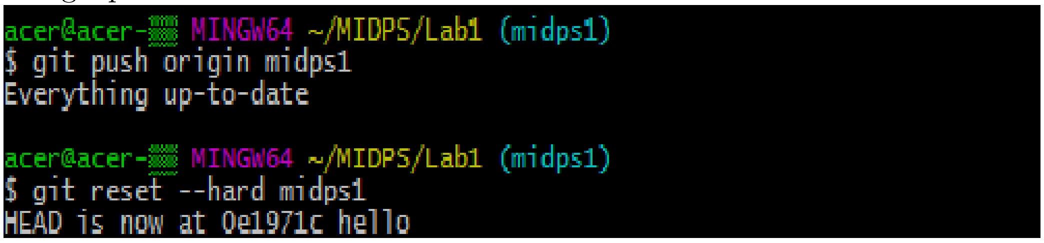

```
acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ git log
commit 0e1971c410fca853f384a7dd2f0b319e4ac7b015
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Wed Feb 8 12:11:10 2017 +0200

    hello

commit 58ac2f4d7e79f19fdc7d76ee562b69c949eaa120
Author: wedwer666 <mitriucmaria@gmail.com>
Date:   Mon Feb 6 11:00:51 2017 +0200

    Update README.md

commit e66f671d42b6559bfde38b4c0a43c416a82e2287
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Mon Feb 6 10:57:37 2017 +0200

    Hello

commit d7cd97eb17c9143f36e4d259c1865c037c7960b1
Author: wedwer666 <mitriucmaria@gmail.com>
Date:   Mon Feb 6 10:22:35 2017 +0200

    Create .gitignore

commit ba57f805297b2e36385dc6827795dc6915994d30
Author: wedwer666 <mitriucmaria@gmail.com>
Date:   Mon Feb 6 09:58:41 2017 +0200

    Create README.md
```

Git reset –soft:



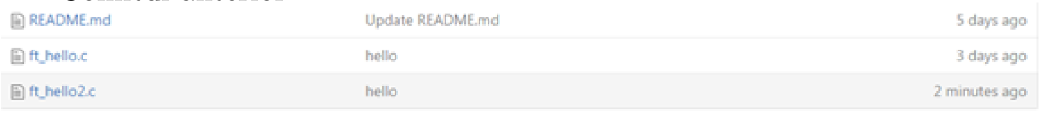
```
acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ git reset midps1

acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ ls
ft_hello.c  README.md

acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ git reset midps1

acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ ls
ft_hello.c  README.md
```

git push



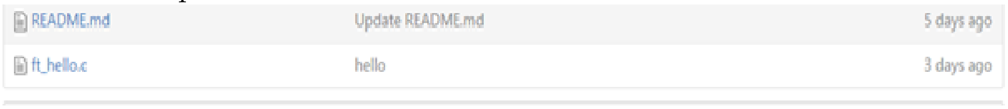
```
acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ git push origin midps1
Everything up-to-date

acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ git reset --hard midps1
HEAD is now at 0e1971c hello
```

Comitul anterior



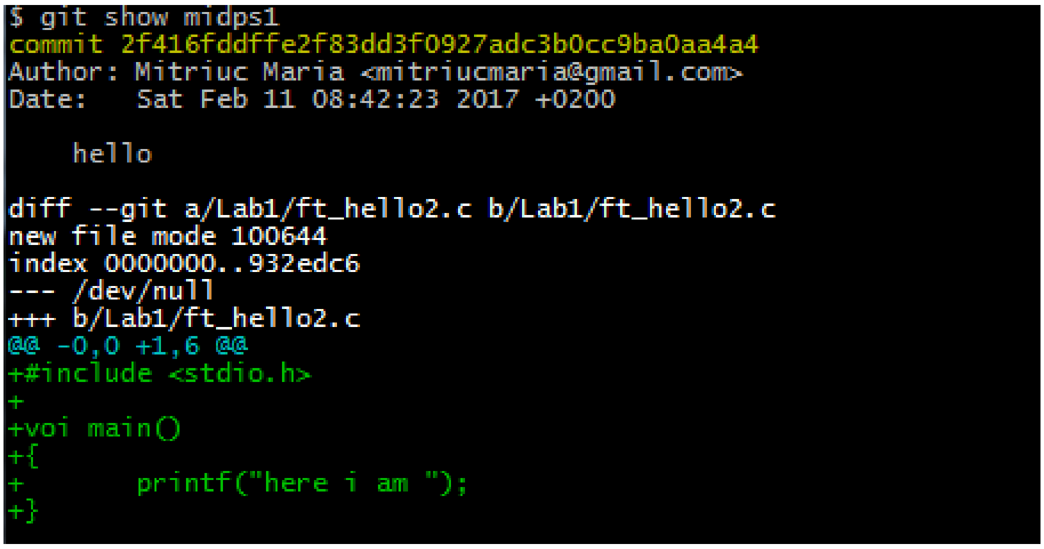
README.md	Update README.md	5 days ago
ft_hello.c	hello	3 days ago
ft_hello2.c	hello	2 minutes ago

Comitul prezent



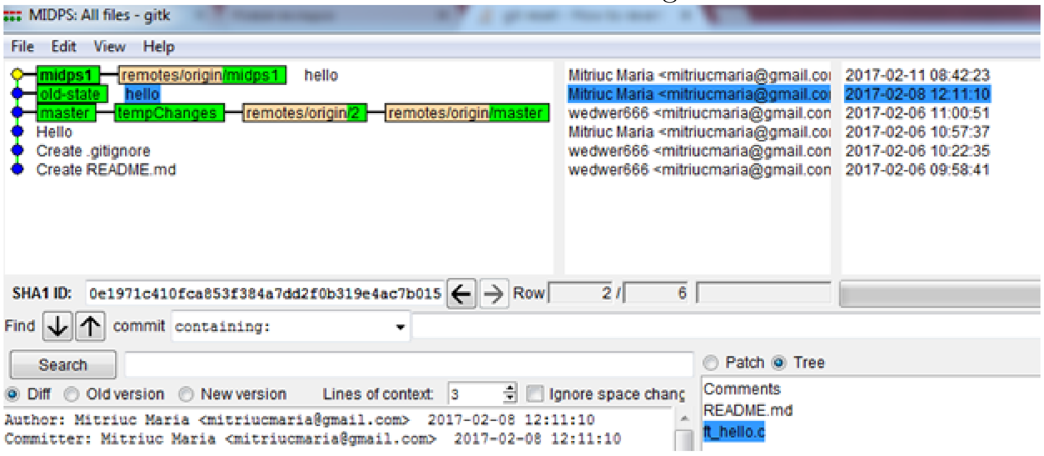
README.md	Update README.md	5 days ago
ft_hello.c	hello	3 days ago

git show



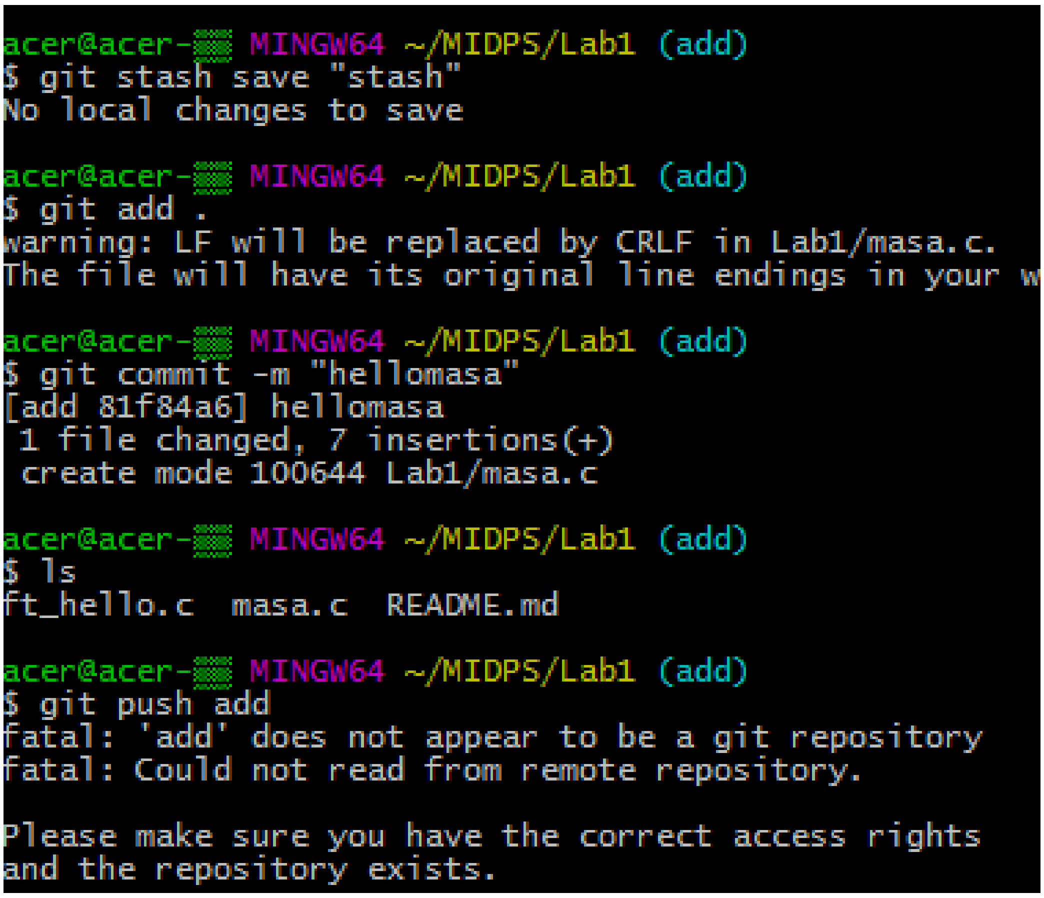
```
$ git show midps1
commit 2f416fddffe2f83dd3f0927adc3b0cc9ba0aa4a4
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Sat Feb 11 08:42:23 2017 +0200

    hello

diff --git a/Lab1/ft_hello2.c b/Lab1/ft_hello2.c
new file mode 100644
index 0000000..932edc6
--- /dev/null
+++ b/Lab1/ft_hello2.c
@@ -0,0 +1,6 @@
+#include <stdio.h>
+
+voi main()
+{
+       printf("here i am ");
+}
```

Analiza rezultatelor folosind instructinunea gitk:



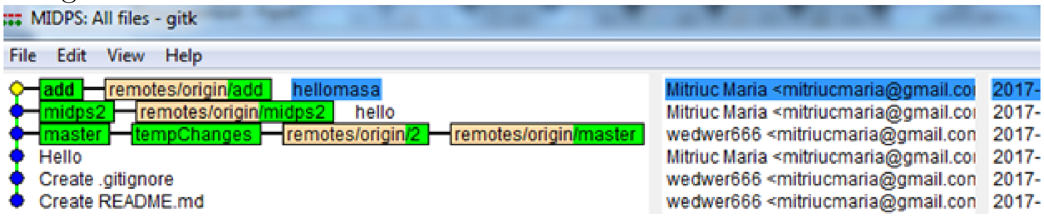
```
MIDPS: All files - gitk
File  Edit  View  Help

midps1 — remotes/origin/midps1  hello          Mitriuc Maria <mitriucmaria@gmail.coi   2017-02-11 08:42:23
old-state  hello                               Mitriuc Maria <mitriucmaria@gmail.coi   2017-02-08 12:11:10
master — tempChanges — remotes/origin/2 — remotes/origin/master
                                               wedwer666 <mitriucmaria@gmail.con       2017-02-06 11:00:51
Hello                                          Mitriuc Maria <mitriucmaria@gmail.coi   2017-02-06 10:57:37
Create .gitignore                              wedwer666 <mitriucmaria@gmail.con       2017-02-06 10:22:35
Create README.md                               wedwer666 <mitriucmaria@gmail.con       2017-02-06 09:58:41

SHA1 ID: 0e1971c410fca853f384a7dd2f0b319e4ac7b015   Row 2 / 6
Find  commit containing:
Search                                         ○ Patch ◉ Tree
◉ Diff ○ Old version ○ New version   Lines of context: 3   ☐ Ignore space chang
Author: Mitriuc Maria <mitriucmaria@gmail.com>  2017-02-08 12:11:10      Comments
Committer: Mitriuc Maria <mitriucmaria@gmail.com>  2017-02-08 12:11:10   README.md
                                                                         ft_hello.c
```

Salvarea temporara a schimbarilor care nu se vor face commit imediat: -git stash:



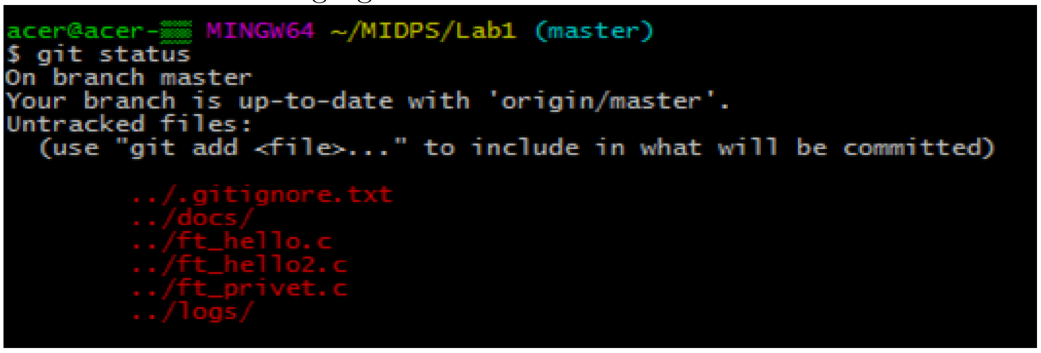
```
acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (add)
$ git stash save "stash"
No local changes to save

acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (add)
$ git add .
warning: LF will be replaced by CRLF in Lab1/masa.c.
The file will have its original line endings in your w

acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (add)
$ git commit -m "hellomasa"
[add 81f84a6] hellomasa
 1 file changed, 7 insertions(+)
 create mode 100644 Lab1/masa.c

acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (add)
$ ls
ft_hello.c  masa.c  README.md

acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (add)
$ git push add
fatal: 'add' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

gitk:



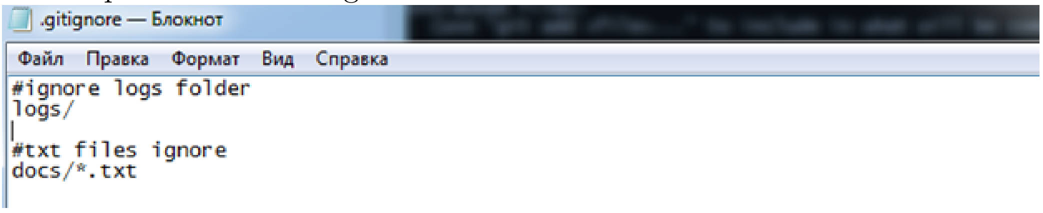
```
MIDPS: All files - gitk
File  Edit  View  Help
add — remotes/origin/add   hellomasa                                   Mitriuc Maria <mitriucmaria@gmail.co   2017-
midps2 — remotes/origin/midps2   hello                                 Mitriuc Maria <mitriucmaria@gmail.co   2017-
master — tempChanges — remotes/origin/2 — remotes/origin/master        wedwer666 <mitriucmaria@gmail.con      2017-
Hello                                                                  Mitriuc Maria <mitriucmaria@gmail.co   2017-
Create .gitignore                                                      wedwer666 <mitriucmaria@gmail.con      2017-
Create README.md                                                       wedwer666 <mitriucmaria@gmail.con      2017-
```

Folosire fisierului .gitignore:



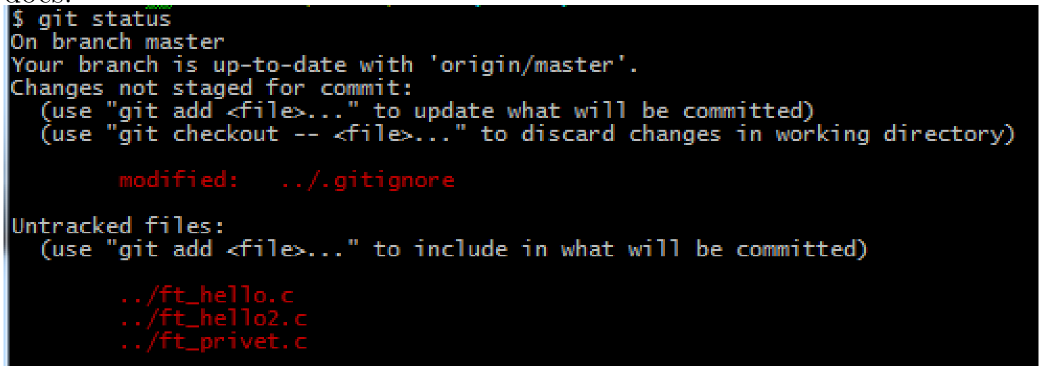
```
acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        ../.gitignore.txt
        ../docs/
        ../ft_hello.c
        ../ft_hello2.c
        ../ft_privet.c
        ../logs/
```

Mapile care o sa fie ignorate:



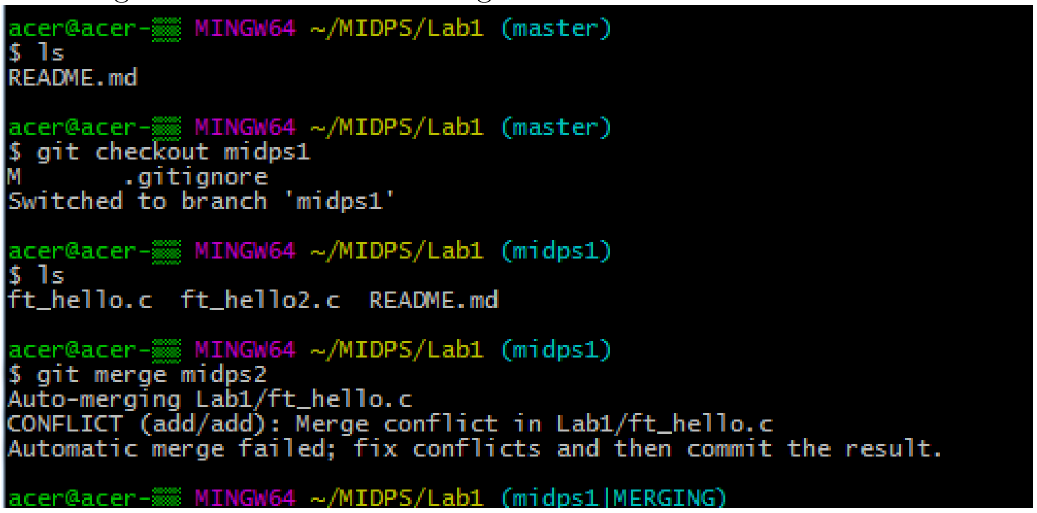
```
.gitignore — Блокнот
Файл  Правка  Формат  Вид  Справка
#ignore logs folder
logs/

#txt files ignore
docs/*.txt
```

Au fost ignorate mapile logs si documentele cu extensia txt din mapa docs:



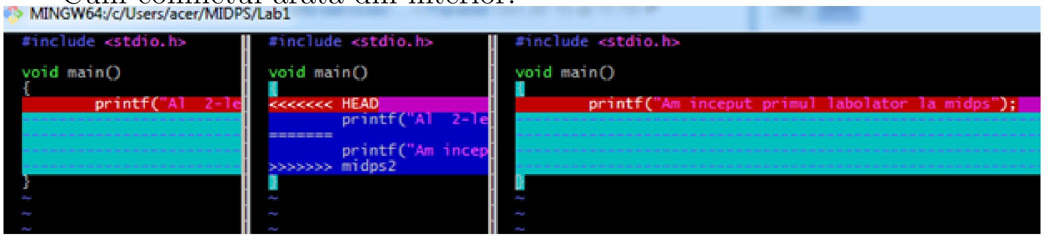
```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   ../.gitignore

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

        ../ft_hello.c
        ../ft_hello2.c
        ../ft_privet.c
```

Merge 2 branch-uri in unul singur + Rezolvarea conflictelor



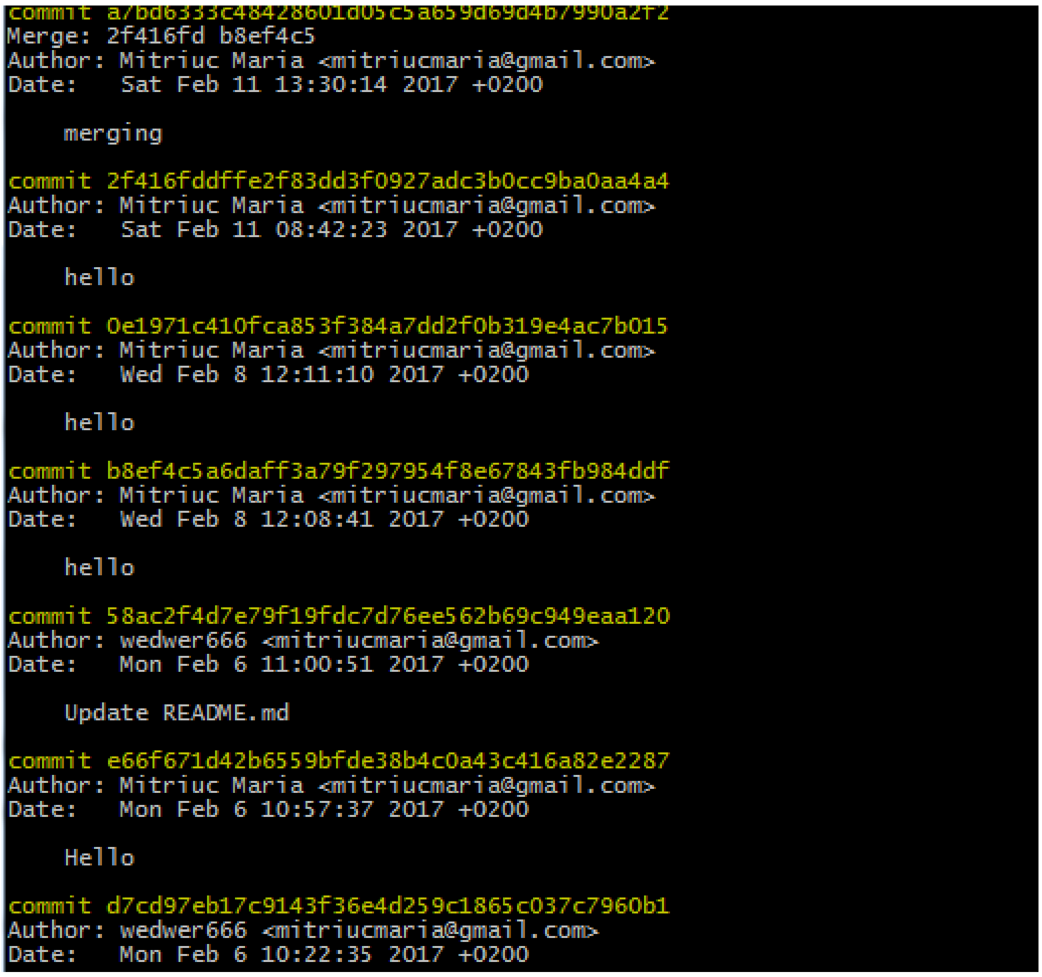
```
acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (master)
$ ls
README.md

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (master)
$ git checkout midps1
M       .gitignore
Switched to branch 'midps1'

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ ls
ft_hello.c  ft_hello2.c  README.md

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1)
$ git merge midps2
Auto-merging Lab1/ft_hello.c
CONFLICT (add/add): Merge conflict in Lab1/ft_hello.c
Automatic merge failed; fix conflicts and then commit the result.

acer@acer-[redacted] MINGW64 ~/MIDPS/Lab1 (midps1|MERGING)
```

Cum conflictul arata din interior:



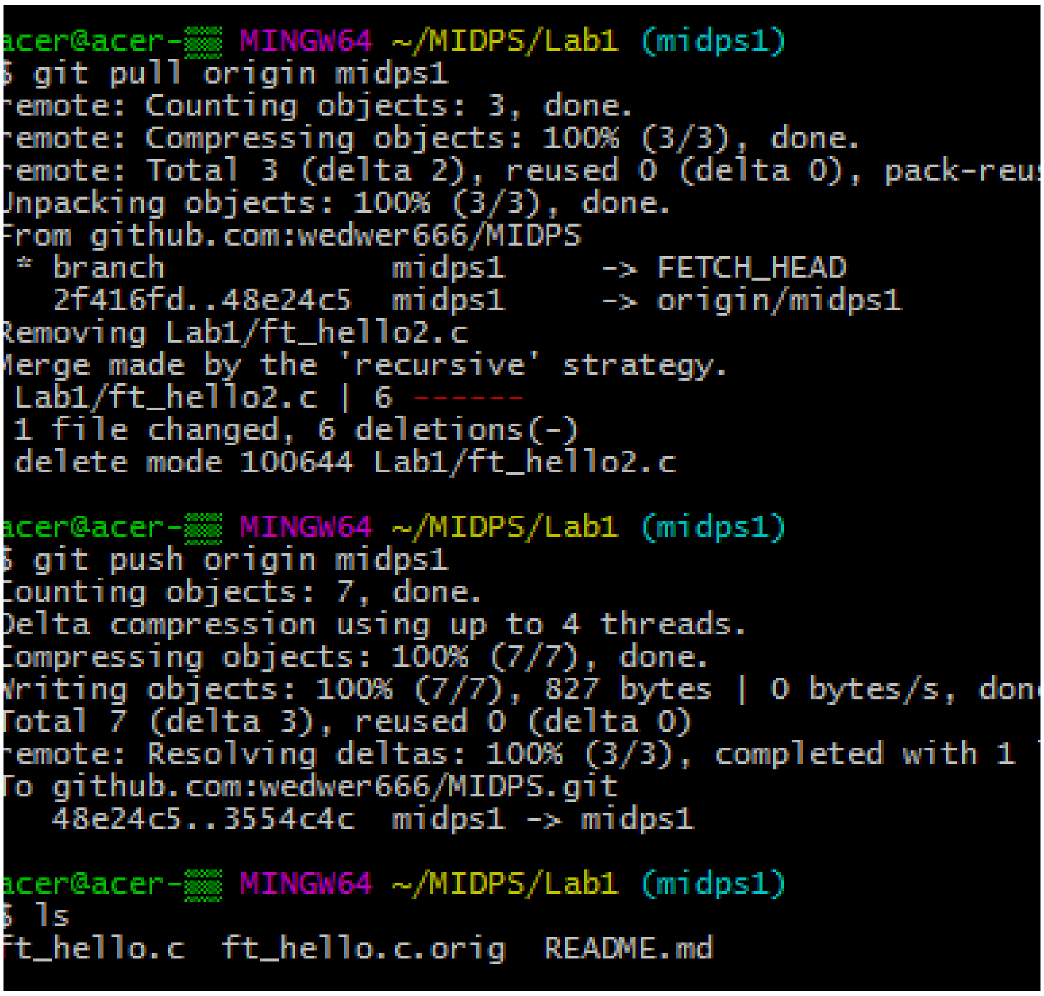
```
MINGW64:/c/Users/acer/MIDPS/Lab1
#include <stdio.h>          | #include <stdio.h>          | #include <stdio.h>
                            |                             |
void main()                 | void main()                 | void main()
{                           | {                           | {
        printf("Al  2-le    | <<<<<<< HEAD                |         printf("Am inceput primul labolator la midps");
                            |         printf("Al  2-le    |
                            | =======                     |
                            |         printf("Am incep    |
                            | >>>>>>> midps2              |
}                           | }                           | }
~                           | ~                           | ~
~                           | ~                           | ~
~                           | ~                           | ~
```

Git log pentru a vedea cum are loc merge:

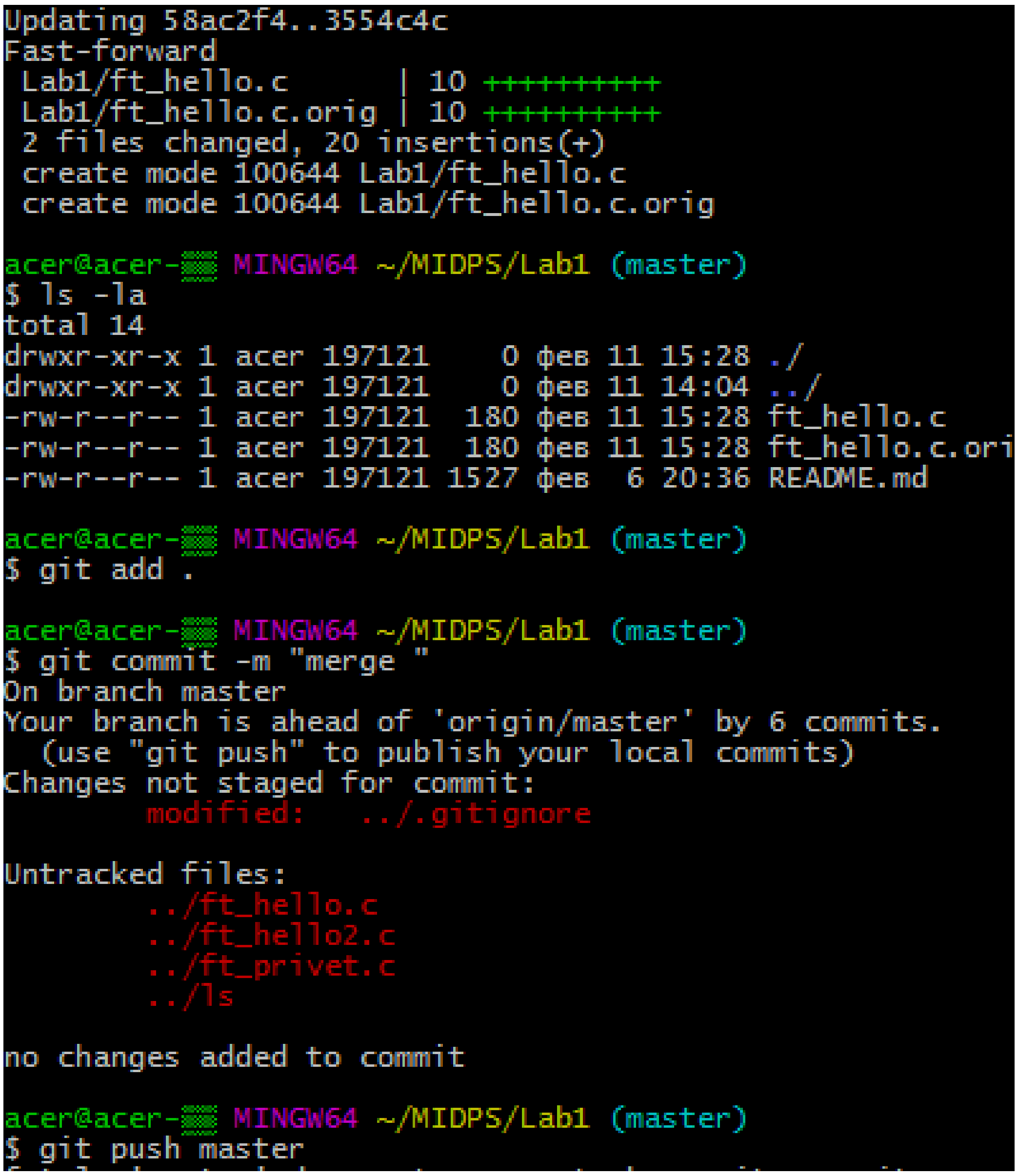


```
commit a7bd6333c48428601d05c5a659d69d4b7990a2f2
Merge: 2f416fd b8ef4c5
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Sat Feb 11 13:30:14 2017 +0200

    merging

commit 2f416fddffe2f83dd3f0927adc3b0cc9ba0aa4a4
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Sat Feb 11 08:42:23 2017 +0200

    hello

commit 0e1971c410fca853f384a7dd2f0b319e4ac7b015
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Wed Feb 8 12:11:10 2017 +0200

    hello

commit b8ef4c5a6daff3a79f297954f8e67843fb984ddf
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Wed Feb 8 12:08:41 2017 +0200

    hello

commit 58ac2f4d7e79f19fdc7d76ee562b69c949eaa120
Author: wedwer666 <mitriucmaria@gmail.com>
Date:   Mon Feb 6 11:00:51 2017 +0200

    Update README.md

commit e66f671d42b6559bfde38b4c0a43c416a82e2287
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Mon Feb 6 10:57:37 2017 +0200

    Hello

commit d7cd97eb17c9143f36e4d259c1865c037c7960b1
Author: wedwer666 <mitriucmaria@gmail.com>
Date:   Mon Feb 6 10:22:35 2017 +0200
```

Folosirea comenzii git pull:

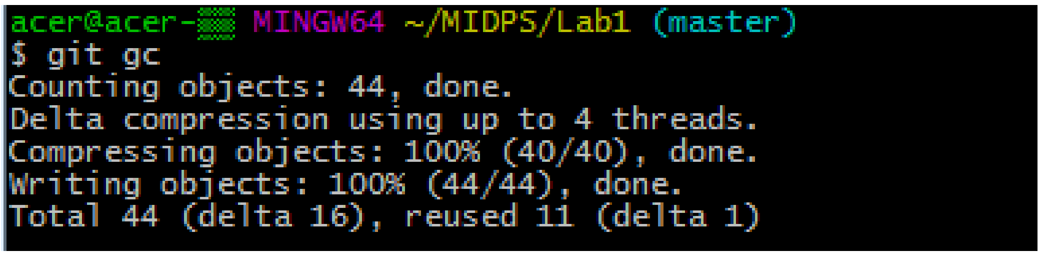


```
acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ git pull origin midps1
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reus
Unpacking objects: 100% (3/3), done.
From github.com:wedwer666/MIDPS
 * branch            midps1     -> FETCH_HEAD
   2f416fd..48e24c5  midps1     -> origin/midps1
Removing Lab1/ft_hello2.c
Merge made by the 'recursive' strategy.
 Lab1/ft_hello2.c | 6 ------
 1 file changed, 6 deletions(-)
 delete mode 100644 Lab1/ft_hello2.c

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ git push origin midps1
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 827 bytes | 0 bytes/s, don
Total 7 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 1
To github.com:wedwer666/MIDPS.git
   48e24c5..3554c4c  midps1 -> midps1

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (midps1)
$ ls
ft_hello.c  ft_hello.c.orig  README.md
```



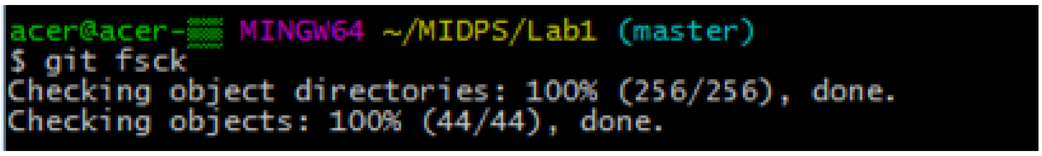
```
Updating 58ac2f4..3554c4c
Fast-forward
 Lab1/ft_hello.c      | 10 ++++++++++
 Lab1/ft_hello.c.orig | 10 ++++++++++
 2 files changed, 20 insertions(+)
 create mode 100644 Lab1/ft_hello.c
 create mode 100644 Lab1/ft_hello.c.orig

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ ls -la
total 14
drwxr-xr-x 1 acer 197121    0 фев 11 15:28 ./
drwxr-xr-x 1 acer 197121    0 фев 11 14:04 ../
-rw-r--r-- 1 acer 197121  180 фев 11 15:28 ft_hello.c
-rw-r--r-- 1 acer 197121  180 фев 11 15:28 ft_hello.c.ori
-rw-r--r-- 1 acer 197121 1527 фев  6 20:36 README.md

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git add .

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git commit -m "merge "
On branch master
Your branch is ahead of 'origin/master' by 6 commits.
  (use "git push" to publish your local commits)
Changes not staged for commit:
        modified:   ../.gitignore

Untracked files:
        ../ft_hello.c
        ../ft_hello2.c
        ../ft_privet.c
        ../ls

no changes added to commit

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git push master
```

Comenzile care trebuie cunoscute



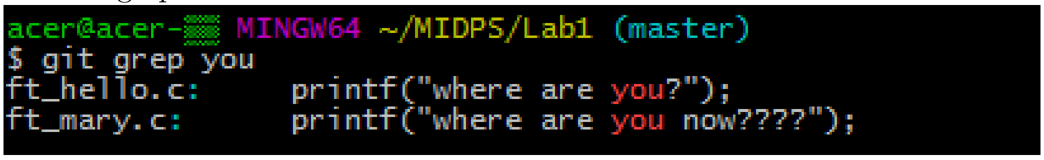
```
acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git gc
Counting objects: 44, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (40/40), done.
Writing objects: 100% (44/44), done.
Total 44 (delta 16), reused 11 (delta 1)
```

Git fsck- verifica validitatea si existent obiectelor in repozitoriu.



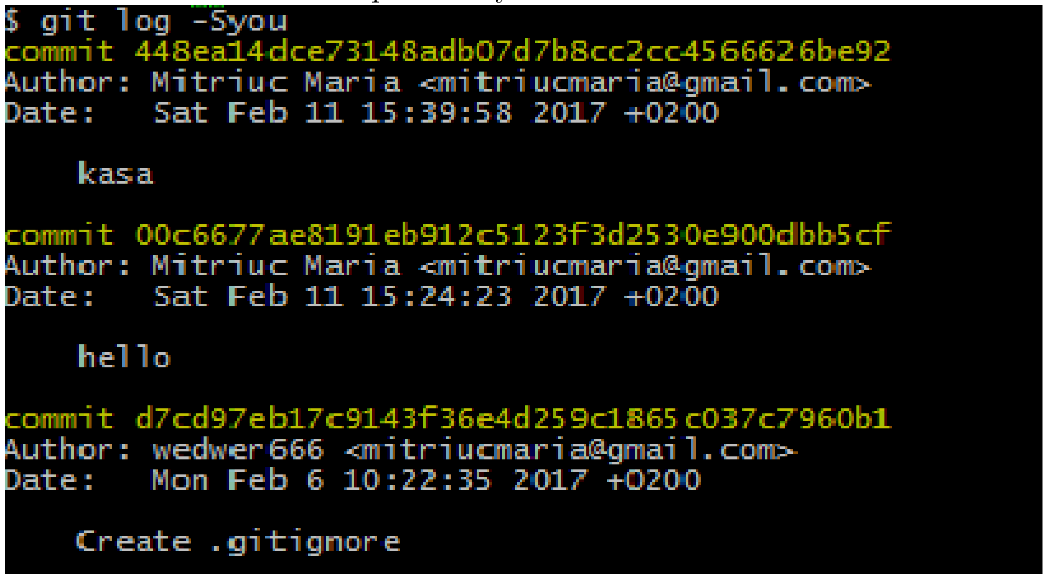
```
acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git fsck
Checking object directories: 100% (256/256), done.
Checking objects: 100% (44/44), done.
```

Git grep:



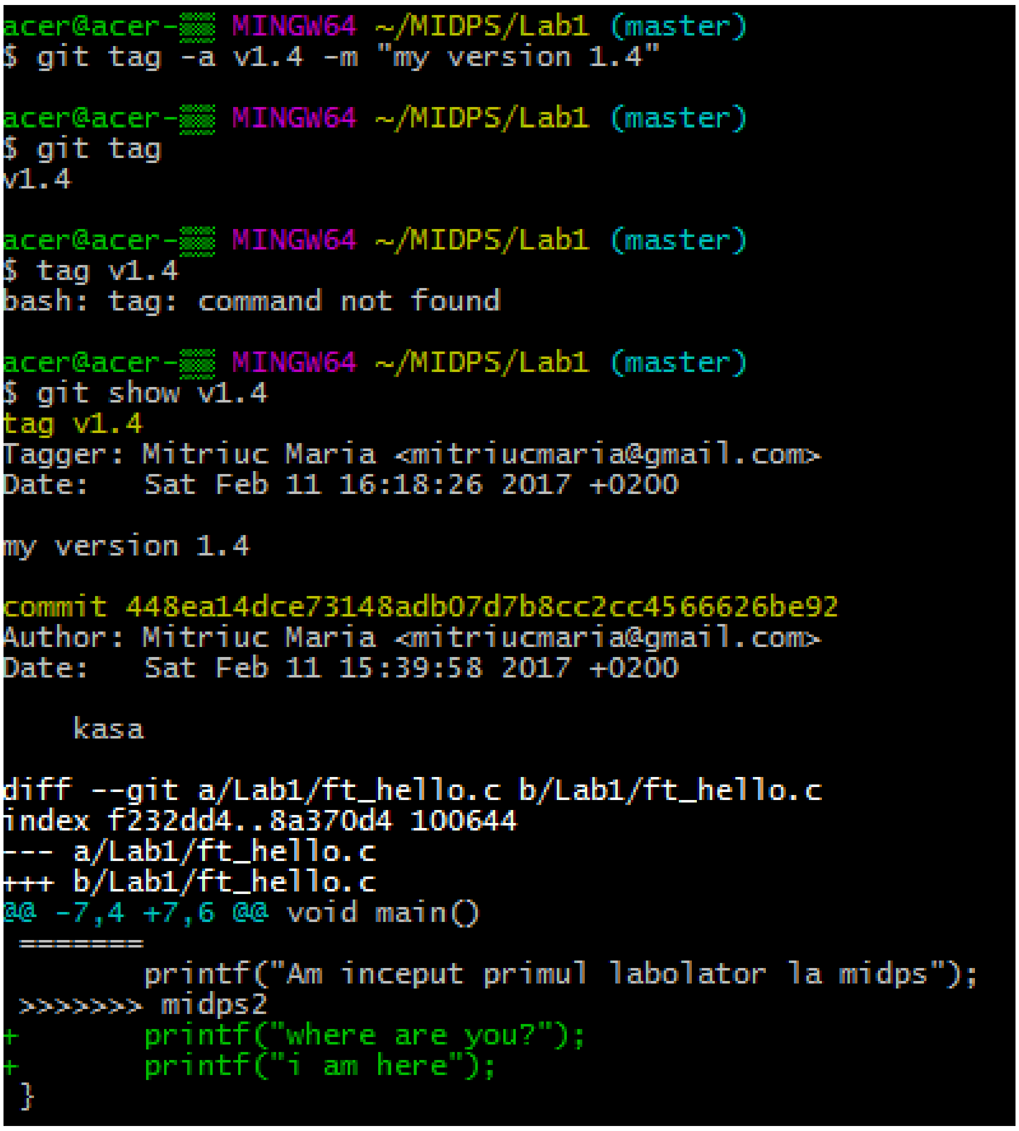
```
acer@acer-▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git grep you
ft_hello.c:     printf("where are you?");
ft_mary.c:      printf("where are you now????");
```

Folosirea unui cuvint specific: ”you”



```
$ git log -Syou
commit 448ea14dce73148adb07d7b8cc2cc4566626be92
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Sat Feb 11 15:39:58 2017 +0200

    kasa

commit 00c6677ae8191eb912c5123f3d2530e900dbb5cf
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Sat Feb 11 15:24:23 2017 +0200

    hello

commit d7cd97eb17c9143f36e4d259c1865c037c7960b1
Author: wedwer666 <mitriucmaria@gmail.com>
Date:   Mon Feb 6 10:22:35 2017 +0200

    Create .gitignore
```

Git tag: Afisarea acestuia:



```
acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git tag -a v1.4 -m "my version 1.4"

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git tag
v1.4

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ tag v1.4
bash: tag: command not found

acer@acer-▒▒ MINGW64 ~/MIDPS/Lab1 (master)
$ git show v1.4
tag v1.4
Tagger: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Sat Feb 11 16:18:26 2017 +0200

my version 1.4

commit 448ea14dce73148adb07d7b8cc2cc4566626be92
Author: Mitriuc Maria <mitriucmaria@gmail.com>
Date:   Sat Feb 11 15:39:58 2017 +0200

    kasa

diff --git a/Lab1/ft_hello.c b/Lab1/ft_hello.c
index f232dd4..8a370d4 100644
--- a/Lab1/ft_hello.c
+++ b/Lab1/ft_hello.c
@@ -7,4 +7,6 @@ void main()
 =======
        printf("Am inceput primul labolator la midps");
 >>>>>>> midps2
+       printf("where are you?");
+       printf("i am here");
 }
```

Concluzie: In acesta lucrare de labolator am facut cunostinta cu principiile de baza ale version control system, avantajele si dezavantajele, am studiat comenzile de baza ale git-ului, am simulat o realizare a proiectului, rezolvarea conflictelor. Implementarea labolatorului a fost realizata prin sistemul de creare a documetelor - LaTeX.

Bibliografie:

1.http://www.debianhelp.co.uk/commands.htm
2.http://www-cs-
students.stanford.edu/ blynn/gitmagic/

3.https://www.youtube.com/playlistlist=PLoonZ8wII66iUm84o7nadL-
oqINzBLk5g

4.https://www.siteground.com/tutorials/git/commands.htm