

Ministerul Educației și Științei
al Republicii Moldova

Universitatea Tehnică a Moldovei

*Facultatea Calculatoare Informatică și Microelectronică
Tehnologii Informationale*

Lucrare de laborator nr.1

la MIDPS

A efectuat:

st.grupeii TI-152
Staver Vasile

A verificat:

Cojanu Irina

Chișinău 2017

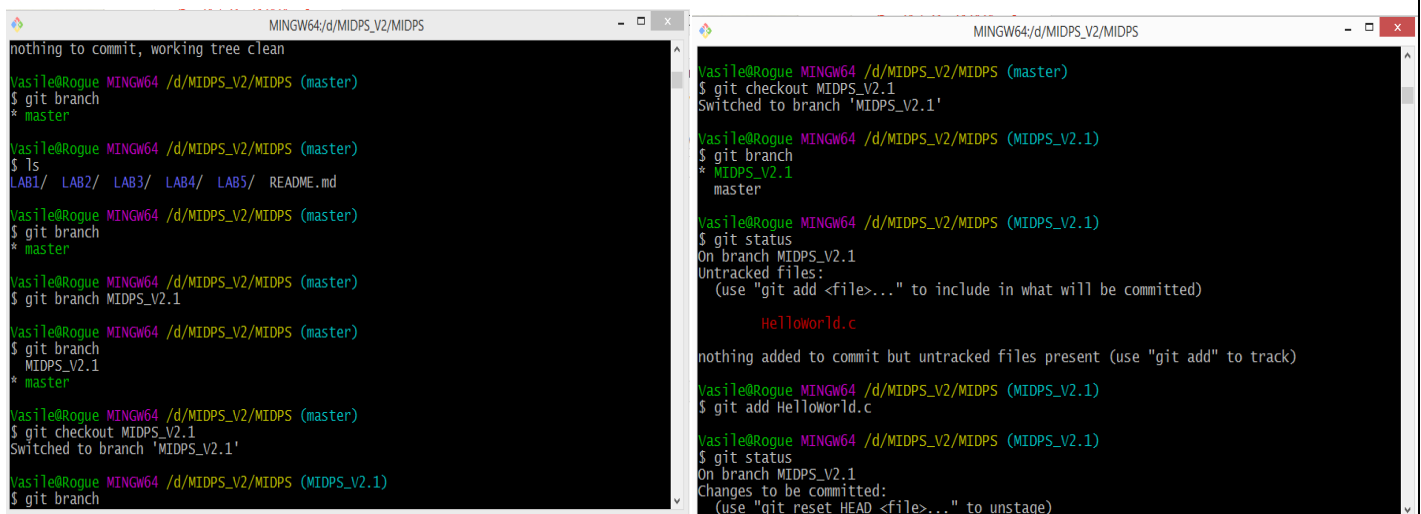
Objective:

- Version Control Systems (git || bitbucket || mercurial || svn)

Laboratory Requirements: Trebuie folosit drept VSC [github](#) sau [bitbucket](#)

- *Basic Level* (nota 5 || 6) :
 - initializeaza un nou repository
 - configureaza-ti VCS
 - crearea 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
 - **crearea branch-urilor (creeaza cel putin 2 branches):**

La crearea branch-urilor am utilizat comanda `git branch branch_name`, am creat doua branch-uri pe care am făcut mai multe commit-uri. Branch-urile le alegeam cu comanda `git checkout`. Pentru a efectua un commit pe un branch este nevoie să creăm mai întâi un fișier în Directiva noastră de lucru apoi prin intermediul `git add` să adăugăm acest fișier în zona virtuală Index, cu ajutorul comenzii `git commit -m` să adăugăm fișierul pe repositoryul nostru local și prin `git push` origin branch să adăugăm fișierul pe branch-ul de pe remote repository (GitHub în cazul nostru)



```
nothing to commit, working tree clean
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (master)
$ git branch
* master

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (master)
$ ls
LAB1/ LAB2/ LAB3/ LAB4/ LAB5/ README.md

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (master)
$ git branch
* master

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (master)
$ git branch MIDPS_V2.1

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (master)
$ git branch
* MIDPS_V2.1
  master

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (master)
$ git checkout MIDPS_V2.1
Switched to branch 'MIDPS_V2.1'

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    HelloWorld.c

nothing added to commit but untracked files present (use "git add" to track)

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git add HelloWorld.c

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
    new file:   HelloWorld.c
```

- *seteaza un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)*

Cu ajutorul comenzii `git branch branch_name --set-upstream-to your_new_remote/branch_name` am setat branch-ul MIDPS_V2 to track a remote origin.

```

MINGW64/d/MIDPS_V2/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git branch -vv
* MIDPS_V2.1 7c728bf Am adaugat un file nou pe branch-ul MIDPS_V2.1
master 96a1fb0 [origin/master: ahead 1] am sters apoi am adaugat fisierele in mapa PS_V2/MIDPS

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git branch -a -vv
* MIDPS_V2.1 7c728bf Am adaugat un file nou pe branch-ul MIDPS_V2.1
n mapa noua MIDPS_V2/MIDPS 96a1fb0 [origin/master: ahead 1] am sters apoi am adaugat fisierele in mapa noua MIDPS_V2/MIDPS
remotes/github/MIDPS_V2.1 7c728bf Am adaugat un file nou pe branch-ul MIDPS_V2.1
remotes/origin/HEAD -> origin/master
remotes/origin/master 9a1d3cf am adaugat fisierele in mape

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git branch -r -vv
github/MIDPS_V2.1 7c728bf Am adaugat un file nou pe branch-ul MIDPS_V2.1
origin/HEAD -> origin/master
origin/master 9a1d3cf am adaugat fisierele in mape

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git remote -v
github git@github.com:Staver-Vasile/MIDPS.git (fetch)
github git@github.com:Staver-Vasile/MIDPS.git (push)
origin git@github.com:Staver-Vasile/MIDPS.git (fetch)

MINGW64/d/MIDPS_V2/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 7 commits.
(use "git push" to publish your local commits)
nothing to commit, working tree clean

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git push origin MIDPS_V2.1
Counting objects: 13, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (13/13), 1.56 KiB | 0 bytes/s, done.
Total 13 (delta 6), reused 0 (delta 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
git staTo github.com:Staver-Vasile/MIDPS.git
7c728bf..d4d321d MIDPS_V2.1 -> MIDPS_V2.1
tu
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 7 commits.
(use "git push" to publish your local commits)
nothing to commit, working tree clean

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$

```

- *reseteaza un branch la commit-ul anterior*

Pentru a reseta un branch la commit-ul anterior am utilizat comanda `git --hard HEAD`

```

MINGW64/d/MIDPS_V2/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git reset --hard HEAD
HEAD is now at 7c728bf Am adaugat un file nou pe branch-ul MIDPS_V2.1

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git log
commit 7c728bfacff541ca9d9f064587bb68db6944794e
Author: Staver Vasile <staver9081@gmail.com>
Date: Sun Feb 26 22:02:56 2017 +0200

    Am adaugat un file nou pe branch-ul MIDPS_V2.1

commit 96a1fb08e406aca42c7ca590ef4f3f13275b99bb
Author: Staver Vasile <staver9081@gmail.com>
Date: Sun Feb 26 21:36:49 2017 +0200

    am sters apoi am adaugat fisierele in mapa noua MIDPS_V2/MIDPS

commit 9a1d3cf043636ab76d9775c2530941a4389ceafa
Author: Staver Vasile <staver9081@gmail.com>
Date: Mon Feb 13 11:07:32 2017 +0200

    am adaugat fisierele in mape

commit 6a74bdd61591d2ea21227583ce69c330d21b0807
Author: bla999 <staver9081@gmail.com>
Date: Mon Feb 13 10:16:26 2017 +0200

```

- *salvarea temporara a schimbarilor care nu se vor face commit imediat.*

Salvarea temporară a schimbărilor presupune situația în care dacă ai făcut careva modificări într-un file și nu le-ai terminat dar dorești să faci altă sarcină, să treci pe alt branch și ca să nu faci commit la un program neterminat utilizezi comanda `git stash` care salvează totul pe un stack. Cu ajutorul comenzii `git stash list` poți să vizualizezi fișierile care sunt în stack iar pentru a reveni la utilizezi comanda `git stash apply`

```

MINGW64/d/MIDPS_V2/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)
nothing to commit, working tree clean

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Hello.txt

nothing added to commit but untracked files present (use "git add" to track)

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git add Hello.txt

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git commit -m "Am adaugat un file Hello.txt"
[MIDPS_V2.1 d07f4a8] Am adaugat un file Hello.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Hello.txt

MINGW64/d/MIDPS_V2/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 3 commits.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
        modified:   Hello.txt

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git stash
Saved working directory and index state WIP on MIDPS_V2.1: d07f4a8 Am adaugat un file Hello.txt
HEAD is now at d07f4a8 Am adaugat un file Hello.txt

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 3 commits.
nothing to commit, working tree clean

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git stash list
stash@0: WIP on MIDPS_V2.1: d07f4a8 Am adaugat un file Hello.txt

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)

MINGW64/d/MIDPS_V2/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 3 commits.
Changes not staged for commit:
  (use "git push" to publish your local commits)
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
        modified:   Hello.txt

no changes added to commit (use "git add" and/or "git commit -a")

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git add Hello.txt

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 3 commits.
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:   Hello.txt

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

```

○ folosirea fisierului .gitignore

Pentru toate „untracked files” sau fişierele la care nu dorim să facem commit şi să apară în repoziitoriul nostru este utilizat fişierul .gitignore care presupune că toate fişierele sau extensiile din acest fişier sunt ignorate. La realizarea acestei sarcini eu am introdus în fişierul gitignore extensia .txt pentru a demonstra aceasta proprietate.

```

.gitignore - Notepad
File Edit Format View Help
# Visual Studio 6 workspace options file
*.opt

# Visual Studio LightSwitch build output
**/*.HTMLClient/GeneratedArtifacts
**/*.DesktopClient/GeneratedArtifacts
**/*.DesktopClient/ModelManifest.xml
**/*.Server/GeneratedArtifacts
**/*.Server/ModelManifest.xml
_Pvt_Extensions

# Paket dependency manager
.paket/paket.exe
paket-files/

# FAKE - F# Make
.fake/

# JetBrains Rider
.idea/
*.sln.inl
*.txt

MINGW64/d/MIDPS_V2/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 6 commits.
nothing to commit, working tree clean

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 6 commits.
Changes not staged for commit:
  (use "git push" to publish your local commits)
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
        modified:   .gitignore

no changes added to commit (use "git add" and/or "git commit -a")

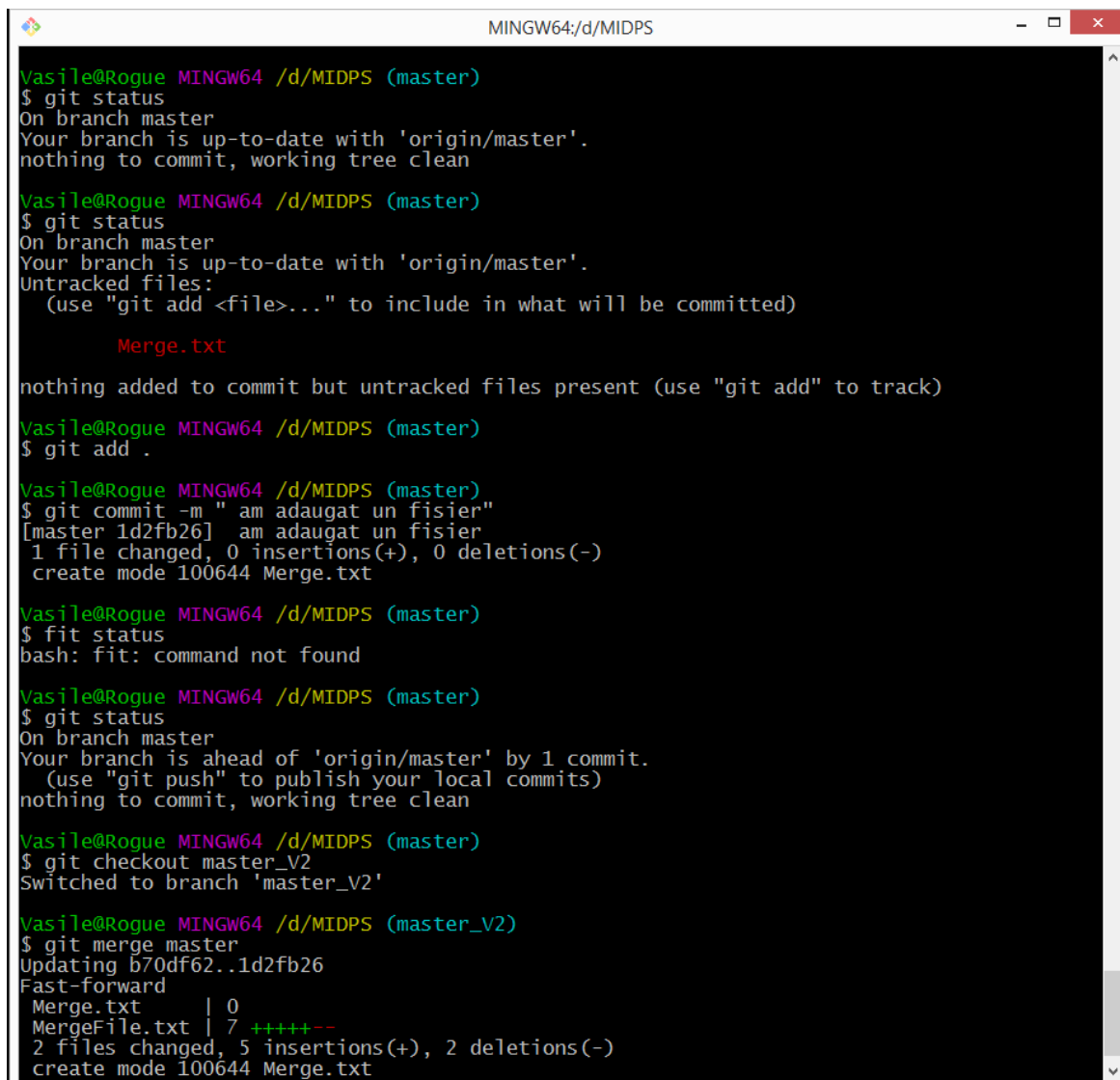
Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git add .

Vasile@Rogue MINGW64 /d/MIDPS_V2/MIDPS (MIDPS_V2.1)
$ git status
On branch MIDPS_V2.1
Your branch is ahead of 'origin/master' by 6 commits.

```

- *merge 2 branches*

Merge-ul este una din cele mai importante funcții în Git care permite să copieze toate fișierele (sau conținutul fișierelor cu același nume) dintr-un branch în altul. Acest lucru permite lucrul mai multor echipe asupra aceluiași program pe branch-uri diferite să fie unit cu ajutorul merge-ului într-un program final. La realizarea acestei sarcini eu doar am făcut merge între două branch-uri identice cu excepția unui singur fișier ceea ce a permis ca fișierul dat să fie copiat în branch-ul în care acesta lipsea.



```
MINGW64:/d/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean

Vasile@Rogue MINGW64 /d/MIDPS (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)

        Merge.txt

nothing added to commit but untracked files present (use "git add" to track)
Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git add .
Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git commit -m " am adaugat un fisier"
[master 1d2fb26] am adaugat un fisier
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Merge.txt

Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git status
bash: git: command not found

Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working tree clean

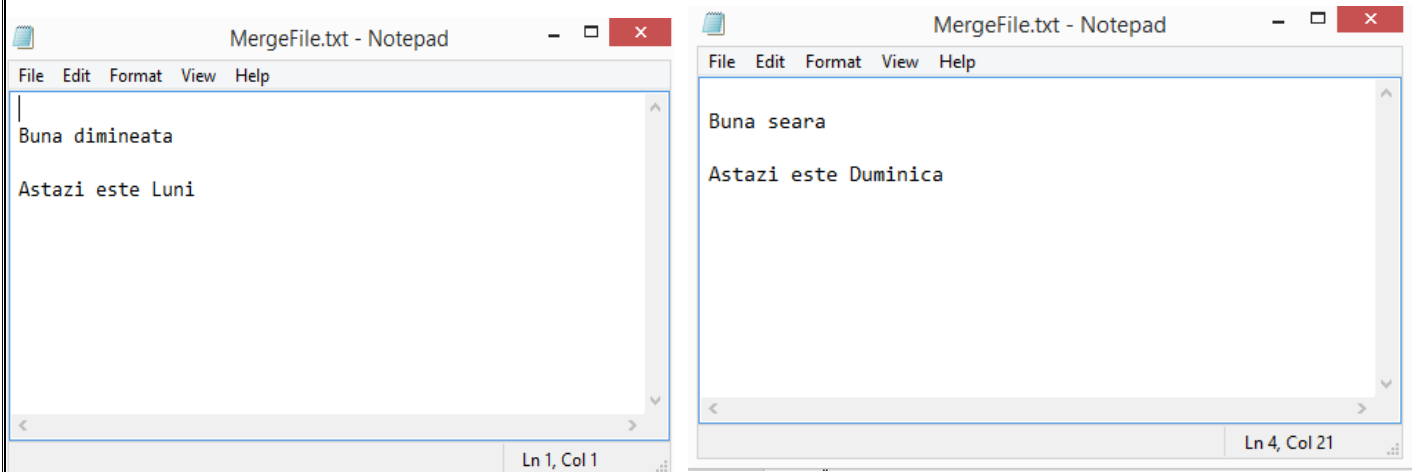
Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git checkout master_V2
Switched to branch 'master_V2'

Vasile@Rogue MINGW64 /d/MIDPS (master_V2)
$ git merge master
Updating b70df62..1d2fb26
Fast-forward
 Merge.txt      | 0
 MergeFile.txt | 7 +++++-
 2 files changed, 5 insertions(+), 2 deletions(-)
create mode 100644 Merge.txt
```

- *rezolvarea conflictelor a 2 branches*

Conflictele între două branch-uri apar atunci când același fișier pe aceleași rânduri este modificat în ambele branch-uri sau un același fișier într-un branch este șters iar în al 2-lea modificat și când are loc merge-ul git-ul nu știe care variantă trebuie să fie aleasă. Eu pentru a crea conflicte am modificat același fișier în ambele branch-uri. Sunt mai multe metode de a rezolva conflictele, eu am utilizat metoda manuală.

Fișierele din cele 2 branch-uri:

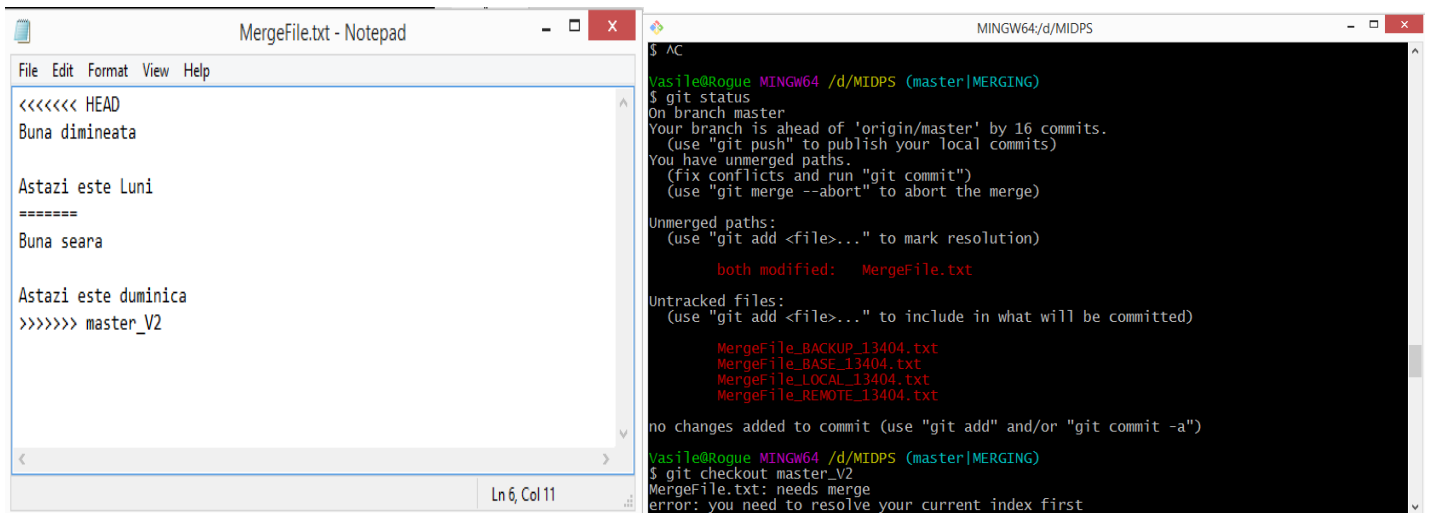


După efectuarea merge-ului:

```
MINGW64:/d/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git merge master_V2
Auto-merging MergeFile.txt
CONFLICT (content): Merge conflict in MergeFile.txt
Automatic merge failed; fix conflicts and then commit the result.

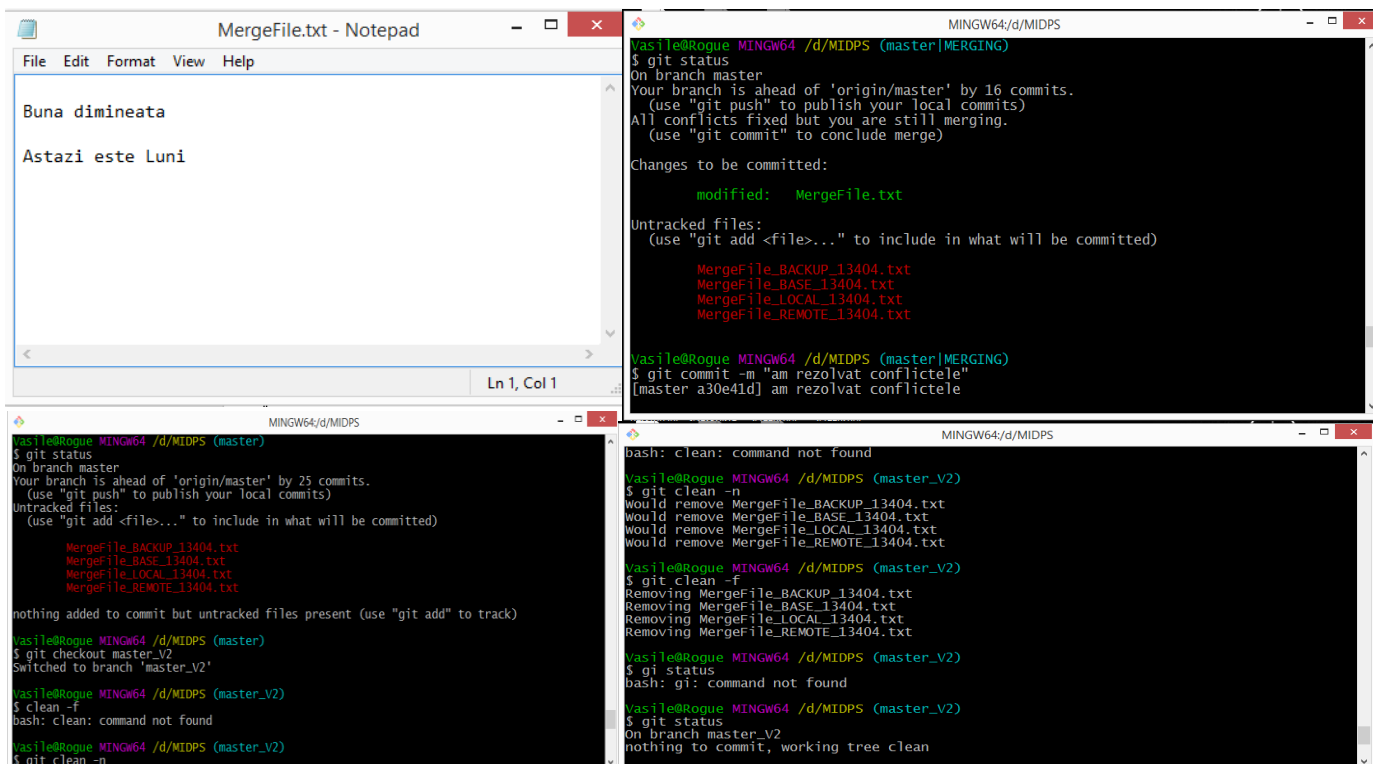
Vasile@Rogue MINGW64 /d/MIDPS (master|MERGING)
$ git mergetool --tool=emerge
Merging:
MergeFile.txt

Normal merge conflict for 'MergeFile.txt':
  {local}: modified file
  {remote}: modified file
The merge tool emerge is not available as 'emacs'
```



În cazul dat pentru soluționarea conflictului trebuie să alegem una din variantele despărțite de „=====” și să ștergem rândurile care au apărut în urma merge-ului.

Astfel obținem fișierul:



```
File Edit Format View Help

Buna dimineata

Astazi este Luni

Ln 1, Col 1

MINGW64/d/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS (master|MERGING)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 16 commits.
(use "git push" to publish your local commits)
All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  modified:   MergeFile.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  MergeFile_BACKUP_13404.txt
  MergeFile_BASE_13404.txt
  MergeFile_LOCAL_13404.txt
  MergeFile_REMOTE_13404.txt

MINGW64/d/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 25 commits.
(use "git push" to publish your local commits)
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  MergeFile_BACKUP_13404.txt
  MergeFile_BASE_13404.txt
  MergeFile_LOCAL_13404.txt
  MergeFile_REMOTE_13404.txt

nothing added to commit but untracked files present (use "git add" to track)

MINGW64/d/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS (master)
$ git checkout master_V2
Switched to branch 'master_V2'

MINGW64/d/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS (master_V2)
$ clean -f
bash: clean: command not found

MINGW64/d/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS (master_V2)
$ git clean -n
bash: git: command not found

MINGW64/d/MIDPS
Vasile@Rogue MINGW64 /d/MIDPS (master_V2)
$ git status
On branch master_V2
nothing to commit, working tree clean
```

Concluzie:

La efectuarea acestei lucrări de laborator eu am făcut cunoștință cu sistemul revision control „Git” și cu GitHub-ul este un serviciu de găzduire web pentru proiecte de dezvoltare a software-ului care utilizează sistemul de control al versiunilor Git. Am înțeles că git-ul reprezintă un instrument extrem de util pentru scrierea și gestionarea programelor largi și realizării versiunilor. La realizarea sarcinilor propuse am studiat proprietățile de creare a repozitoriilor locale cât și „remote”. Am analizat comenzile de bază a git-ului așa ca *add* și *commit*. Am învățat așa termeni ca „Working directory” care reprezintă mapa în care noi lucrăm, „Index” –ul care reprezintă o zonă virtuală în care noi încărcăm toate fișierile și modificările respective ca mai apoi prin „commit” să încărcăm aceste fișiere în repositoryul nostru local. Deasemenea exista și „Remote repository” care în cazul nostru reprezintă Github-ul astfel prin intermediul unor comenzi ca *push*, *fetch*, *pull* putem comunica între aceste repozitorii. Deasemenea am studiat branch-urile care permit unor persoane diferite să dezvolte același soft independent unii de alții și ținând cont de toate modificările anterioare și să întrunească într-un final prin așa comandă ca *merge* toate proprietățile unui program efectuat anterior de grupuri de oameni diferiți. Deasemenea am văzut și am implementat conflictele care pot apărea între 2 branch-uri și modalitățile de soluționare a acestora.

Bibliografie:

- google.com
- youtube.com
- <https://git-scm.com/book/en/v2/>