**SWE 525 GIT Version Control Mid Term Exam 04/10/2016**

**ITU Student Name: Murali Krishna Gogula**

**ITU Student ID: 90584**

**Note: Please note that exam will be passed through “turn it in” All plagiarism cases will be reported to ITU administrators so don’t copy and collaborate.**

Prof please check my **Midterm** repo in my github account, because I am going to attach this doc over there along with few others also.

Thank you.

Total 18 pages

* 1. List out the key difference between a centralized version control system and distributed version control system. List down any two centralized version control system and 2 distributed version control system. What are the advantages of git VCS over other VCS. What are the different states of a file in the Git VCS.

Answer:

**Differences**

In a Centralized Version Control System there will be one Central copy or Repository of the project to which all the programmers involved submit their changes and commit to it. Other programmers can also pull the changes made and submit their own changes to the central copy. It will automatically update all the changes made.

* In Distributed Version Control system there is no need to rely on a single central server instead that central server can be cloned and copies are made of the repository and each developer will be able to submit the changes to their own respective clones of that central repository. The metadata of the original will be stored in each clone made. We can perform pull and push operations in this VCS. Committing new change sets can be done locally without anyone else seeing them.

**Examples:**

Examples for Centralized version control systems: CVS, Subversion and Perforce.

Examples for Distributed Version Control System: Mercurial, Git and Bazaar.

**Advantages:**

Git has an advantage of cloning the central repository into several repositories. New changes can be committed locally without anyone else seeing them. All the change sets can be sent in a group using push command.

Each programmer can share changes made to the repository by collaborating with other users to get their opinion before sharing with others.

**Different States of a File**:

There are three different states of a file in Git they are

* Working directory
* Staging area
* Git directory (repository)
* 2. Perform following Remote Repository operations. Please put commands, screenshots and other information where needed. More detailed oriented response will get higher points. Give me your github repo link also with following steps added.

1. Create a repository on Github and configure your local repo to point to the remote repository
2. Perform some operation like add, remove, modify and finally push your changes to the remote repository
3. Pull the latest changes from the repository to get the updates from others in to your local repo and merge the changes
4. Try fetching the changes and perform the merge to get the difference between the pull and the merge command
5. Perform some changes and before committing the changes, stash your changes and then pull the changes and finally apply the changes to understand how stashing works
6. Create a feature branch and do some file operations in the branch and commit the changes to the branch
7. Merge the changes using the rebase command and finally perform a safe deletion of the feature branch
8. Create another feature branch and this time after committing the changes to the feature branch, merge the changes using fast forward merge and then delete the feature branch

Ans: Here I have listed only commands because of the size. For this question I will attach another file in my Midterm repo folder along with this.

  git init

184 git clone https://github.com/muralicloud/muralione.git

185 cd myrepo

186 vi saiteja12.txt

187 git add saiteja12.txt

188 git commit -m "fifhiifuh"

189 git push origin master

190 git push -u origin master

191 git push

192 git push -u origin master

193 cd ..

194 pwd

195 git status

196 cd myrepo

197 ls

198 git status

199 git add saiteja12.txt

200 ls

201 git status

202 git commit -m "lsdhfs"

203 cd muralione

204 mkdir muralione

205 cd muralione

206 cd..

207 cd ..

208 cd ..

209 mkdir muralione

210 cd muralione

211 ls

212 git add README.md

213 git status

214 git commit -m "fusdf"

215 vi meet

216 git add meet

217 git commit -m "meet"

218 git push -u origin master

219 git pull

220 vi saiii

221 git add\*

222 git add --all

223 git commit -m "ajd"

224 git push

225 git pull

226 git status

227 ls

228 exit

229 git init

230 git clone https://github.com/muralicloud/saitejak.git

231 vi saai.html

232 git status

233 git add saai.html

234 git status

235 git commit -m "knjsdfkdsj"

236 git status

237 mkdir saitejak

238 ls

239 cd saitejak

240 vi saaai.txt

241 git add saai.txt

242 git add saaai.txt

243 git commit -m ",jhfksdhf"

244 git status

245 git log

246 git log --stat

247 git log --graph

248 git push

249 git pull

250 ls

251 vi kamuni

252 git statu

253 git status

254 git commit -m "sdfhk"

255 git push

256 git push

257 ls

258 vi kamuni

259 git add kamuni

260

261 git push

262 ls

263 git push

264 git init

265 vi fss.txt

266 git add --all

267 git commit -m ",jfjdsfn"

268 git push

269 git init

270 mkdir murali

271 cd murali

272 ls

273 ls -l

274 git clone https://github.com/muralicloud/gogula.git

275 ls

276 ls -l

277 vi hhh.txt

278 ls

279 git add hhh.txt

280 git status

281 git commit -m "first"

282 git status

283 git push

284 git push origin master

285 cd ..

286 ls

287 cd murali

288 ls

289 cd hhh.txt

290 git push hhh.txt

291 git pull

292 git clone

293 git clone https://github.com/muralicloud/gogula.git

294 ls

295 vi README.md

296 git add\*

297 git add --all

298 git add \*

299 git commit -m "murali"

300 git status

301 git add README.md

302 git status

303 git commit -m "gfgjng"

304 git push

305 git push https://github.com/muralicloud/gogula.git

306 ls

307 git commit -m "dfhbdb"

308 git pull

309 git pull https://github.com/muralicloud/gogula.git

310 ls

311 cd gogula

312 ls

313 git push

314 vi krish.txt

315 git add krish.txt

316 git commit -m "kkfbbhf"

317 git push

318 git init

319 vi mmmm.html

320 git add \*

321 ls

322 git commit -m "murali"

323 git push

324 git init

325 git push

326 git pull

327 git push

328 clear

329 cd ..

330 cd ..

331 clear

332 git init

333 git clone https://github.com/muralicloud/Protek1.git

334 git status

335 ls

336 ;s

337 ls

338 cd protek1

339 git init

340 vi batch1

341 vi batch1.html

342 git add --all

343 git status

344 git commit -m "first commit"

345 git status

346 git pull

347 git push

348 vi

349 git --version

350 git --config

351 git --branch

352 git branch

353 cd ..

354 cd ..

355 cd ..

356 git checkout -b

357 git checkout -b master

358 git checkout master

359 git branch master

360 cd master

361 ls

362 cd users

363 cd c

364 cd users

365 cd branch master

366 ccd master

367 cd master

368 ls

369 git branch

370 clear

371 git branch Pro-Tek

372 cd ..

373 ls

374 cd userss

375 cd users

376 clear

377 git branch

378 git checkout -b master

379 git checkout

380 cd master

381 pwd

382 cd muralikrishna

383 git init

384 git clone https://github.com/muralicloud/Cloud4.git

385 git status

386 vi cloud3

387 vi cloud3.txt

388 git add cloud3.txt

389 ls

390 cd cloud4

391 ls

392 vi cloud3.txt

393 git add -all

394 git add --all

395 ls

396 git status

397 vi cloud3.txt

398 git status

399 git diff cloud3.txt

400 git commit -m "first commit"

401 git status

402 git pull

403 git push

404 ls

405 git pull

406 ls

407 git log

408 git stat

409 git graph

410 git log graph

411 git init

412 git clone https://github.com/sait310/cloudrep.git

413 ls

414 ls -al

415 ls -l

416 cd ..

417 ls

418 cd cloudrep

419 cd sai310/cloudrep

420 cd cloud4

421 git clone

422 git clone

423 git clone https://github.com/sait310/cloudrep.git

424 git pull

425 cd ..

426 ls

427 git init

428 git clone https://github.com/sait310/cloudrep.git

429 ls

430 cd cloudrep

431 vi kamuni.txt

432 git add --all

433 git commit -m "dfdfn"

434 git push

435 git branch

436 git saiteja

437 git branch saiteja

438 git branch

439 git checkout

440 git checkout -b

441 git checkout saiteja

442 git -b checkout

443 git checkout

444 git checkout master

445 git shortlog

446 git stash

447 git status

448 git checkout branch master

449 git branch

450 git branch master

451 git checkout -b master

452 git branch checkout -b master

453 git branch checkout master

454 git branch chekout newone

455 git branch

456 git checkout -b masster

457 git checkout -b master

458 exit

459 git checkout branch master

460 ls

461 cd..

462 cd ..

463 ls

464 git branch

465 cd master

466 ls -l

467 git reset

468 git branch master

469 exit

470 cd master

471 ls

472 git branch

473 git checkout murali

474 git checkout master

475 clear

476 ls

477 vi saai.html

478 git add \*

479 git rm -rf saai.html

480 git status

481 mkdir nrepo

482 cd nrepo

483 git clone https://github.com/muralicloud

484 git clone https://github.com/muralicloud/nrepo.git

485 git init

486 ls

487 vi git.html

488 git add \*

489 git status

490 git commit -m "git file"

491 git statu

492 git status

493 git pull

494 git push

495 git init

496 ls

497 cd ..

498 git init

499 git clone https://github.com/muralicloud/nrepo.git

500 exit

501 git init

502 git clone https://github.com/muralicloud/Midterm.git

503 git init

504 ls

505 cd Midterm

506 vi midterm.html

507 git add \*

508 git commit -m "first comit"

509 git status

510 git pull

511 git push

512 git status

513 ls

514 git pull

515 git status

516 ls

517 vi midterm.html

518 git add \*

519 git commit -m "updated"

520 git status

521 git diff

522 git diff midterm.html

523 git push

524 vi midterm.txt

525 git diff

526 git add \*

527 git commit -m "updated"

528 git push

529 git pull

530 ls

531 git rm midterm.txt

532 ls

533 git commit -m "fgjg"

534 git status

535 git push

536 git pull

537 ls

538 cd ..

539 ls

540 git init

541 ls

542 git clone https://github.com/muralicloud/Others.git

543 ls

544 cd Others

545 vi test.html

546 git add \*

547 git commit -m "new file"

548 git status

549 git push

550 git status

551 git pull

552 cd ..

553 ls

554 cd Midterm

555 ls

556 git merge Others

557 git merge "Others"

558 git init

559 git pull https://github.com/muralicloud/Others.git

560 ls

561 git fetch

562 ls

563 git diff

564 git fetch origin

565 vi newone.txt

566 git add \*

567 git status

568 git stash save "newone.txt"

569 git status

570 git stash

571 git stash list

572 git stash pop

573 git stash drop stash

574 git stash show stash

575 git stash drop stash@{2}

576 git commit -m "after stash"

577 ls

578 git stash list

579 git push

580 git init

581 git branch -b midterm

582 git branch midterm

583 git checkout branch -b midterm

584 git checkout -b midterm

585 git checkout -b exam

586 ls

587 cd ..

588 ls

589 git branch

590 git checkout -b midterm

591 ls

592 git brranch

593 git branch

594 git init

595 git clone https://github.com/muralicloud/newdoc.txt.git

596 l

597 ls

598 cd newdoc.txt

599 vi newdocc.txt

600 git add \*

601 git ommit -m "flfnl"

602 git commit -m "flfnl"

603 git push

604 git push

605 git status

606 git init

607 git clone https://github.com/muralicloud/newdoc.txt.git

608 vi new.html

609 git add \*

610 git commit -m "dfdnd"

611 git push

612 git status

613 ls

614 git branch midterm

615 git checkout -b midterm

616 cd ..

617 git branch

618 mkdir newrepo

619 cd newrepo

620 ls

621 cd ..

622 ls

623 git clone https://github.com/muralicloud/newrepo.git

624 cd newrepo

625 ls

626 vi first.txt

627 git add \*

628 git commit -m "jfkjsf"

629 cd ..

630 git clone https://github.com/muralicloud/newrepo.git

631 git clone https://github.com/muralicloud/newrepo.git

632 git clone https://github.com/muralicloud/repo.git

633 ls

634 cd repo

635 git branch

636 git create branch midterm

637 git checkout -b midterm

638 vi first.txt

639 git add \*

640 git commit -m "gngsnj"

641 git push

642 git status

643 ls

644 git push

645 git push origin

646 git push origin midterm

647 git pull

648 git pull origin midterm

649 git checkout midterm

650 git rebase master

651 git checkout master

652 git rebase midterm

653 git checkout midterm

654 git checkout master

655 git merge midterm

656 ls

657 git checkout midterm

658 ls

659 git checkout master

660 ls

661 git branch --merged

662 git midterm --merged

663 git log

664 git rm -rf midterm

665 git rm branch "midterm"

666 git delete -b midterm

667 git branch

668 git rm -b midterm

669 cd ..

670 git delete -b midterm

671 git checkout master

672 git delete -d midterm

673 git checkout midterm

674 git branch -d comment\_broken\_links\_101026

675 git branch -d

676 git branch -d midterm

677 git checkout master

678 git branch -d midterm

679 git history

680 git reflog

681 git reflog

682 history

2. Perform following branching and merging operations. Please put commands, screenshots and other information where needed. More detailed oriented response will get higher points. Give me your github repo link also with following steps added.

1. Create a local branch using git checkout -b branchname command

A) muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout -b mehta

M murali/gogula

D saai.html

Switched to a new branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

1. Observe the difference by doing some file operations and switch back to the master  branch and see if you can see the changes done on the branch

A) muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout -b mehta

M murali/gogula

D saai.html

Switched to a new branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

$ ^C

muralikrishna@Shrita MINGW64 ~ (mehta)

$ vi murali.txt

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git add \*

fatal: Not a git repository: AppData/Local/GitHub/IgnoreTemplates\_a66c3719071da6d865a984bb8d6bfb5bcd775ec8/../.git/modules/gitignore

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git commit

Aborting commit due to empty commit message.

muralikrishna@Shrita MINGW64 ~ (mehta)

$ mkdir test

muralikrishna@Shrita MINGW64 ~ (mehta)

$ cd test

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ vi murali.txt

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git add \*

warning: LF will be replaced by CRLF in test/murali.txt.

The file will have its original line endings in your working directory.

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

D saai.html

A test/murali.txt

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test (master)

$ ls

murali.txt

1. Now switch back to the branch and commit the changes and switch to master  branch. Now see if you can still see the changes in the master branch
2. Now switch back to the branch name and stash the changes and apply the changes  to the master branch by switching to the master branch
3. Try merging the changes from the branch to the master branch using all the three  merge strategies and then view the git log
4. Push the local branch to the remote repository and see if the branch is present on the  remote repository - Github

A) muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout -b mehta

M murali/gogula

D saai.html

Switched to a new branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

$ ^C

muralikrishna@Shrita MINGW64 ~ (mehta)

$ vi murali.txt

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git add \*

fatal: Not a git repository: AppData/Local/GitHub/IgnoreTemplates\_a66c3719071da6d865a984bb8d6bfb5bcd775ec8/../.git/modules/gitignore

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git commit

Aborting commit due to empty commit message.

muralikrishna@Shrita MINGW64 ~ (mehta)

$ mkdir test

muralikrishna@Shrita MINGW64 ~ (mehta)

$ cd test

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ vi murali.txt

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git add \*

warning: LF will be replaced by CRLF in test/murali.txt.

The file will have its original line endings in your working directory.

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

D saai.html

A test/murali.txt

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test (master)

$ ls

murali.txt

muralikrishna@Shrita MINGW64 ~/test (master)

$ ^C

muralikrishna@Shrita MINGW64 ~/test (master)

$ git checkout mehta

M murali/gogula

D saai.html

A test/murali.txt

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ ls

murali.txt

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ vi newmurali.html

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git add \*

warning: LF will be replaced by CRLF in test/newmurali.html.

The file will have its original line endings in your working directory.

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

D saai.html

A test/murali.txt

A test/newmurali.html

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test (master)

$ ls

murali.txt newmurali.html

muralikrishna@Shrita MINGW64 ~/test (master)

$ git checkout mehta

M murali/gogula

D saai.html

A test/murali.txt

A test/newmurali.html

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git commit -m "dsglsngs"

[mehta warning: LF will be replaced by CRLF in test/murali.txt.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in test/newmurali.html.

The file will have its original line endings in your working directory.

cadff03] dsglsngs

warning: LF will be replaced by CRLF in test/murali.txt.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in test/newmurali.html.

The file will have its original line endings in your working directory.

3 files changed, 2 insertions(+), 1 deletion(-)

delete mode 100644 saai.html

create mode 100644 test/murali.txt

create mode 100644 test/newmurali.html

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test

$ ls

ls: cannot open directory .: Permission denied

muralikrishna@Shrita MINGW64 ~/test

$ git checkout master

fatal: Unable to read current working directory: No error

muralikrishna@Shrita MINGW64 ~/test

$ git checkout mehta

fatal: Unable to read current working directory: No error

muralikrishna@Shrita MINGW64 ~/test

$ cd ..

muralikrishna@Shrita MINGW64 ~ (master)

$ cd test

bash: cd: test: No such file or directory

muralikrishna@Shrita MINGW64 ~ (master)

$ git branch

checkout

chekout

masster

\* master

mehta

murali

newone

muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout mehta

M murali/gogula

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

$ ls

3D Objects/ IntelGraphicsProfiles/ newrepo/ protek1.html

AppData/ ITU-/ NOTICE README.md

Application Data@ ituday2/ nrepo/ Recent@

bin/ kuldip/ NTUSER.DAT RELEASE-NOTES

BullseyeCoverageError.txt lib/ ntuser.dat.LOG1 RemoteSystemsTempFiles/

class1/ libgit2/ ntuser.dat.LOG2 repo/

cloud3 LICENSE NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.0.regtrans-ms RUNNING.txt

Cloud4/ Links/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.1.regtrans-ms saiteja/

cloudrep/ Local Settings@ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.2.regtrans-ms saitejak/

conf/ logs/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.blf sample.txt

Contacts/ md NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TM.blf Saved Games/

Cookies@ Midterm/ NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000001.regtrans-ms Searches/

Creative Cloud Files/ murali/ NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000002.regtrans-ms SendTo@

Desktop/ murali.txt ntuser.ini Start Menu@

Documents/ muralione/ OneDrive/ temp/

Downloads/ Music/ Others/ Templates@

Favorites/ My Documents@ Pictures/ test/

fdfd my\_project/ practise/ Tracing/

file1.txt mycode/ practise.html Videos/

first.js/ myrepo/ PrintHood@ VirtualBox VMs/

git/ NetHood@ project/ webapps/

gitday2/ newdoc.txt/ protek/ work/

index.html newproject/ Protek1/ workspace/

muralikrishna@Shrita MINGW64 ~ (mehta)

$ cd test

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ ls

murali.txt newmurali.html

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test

$ cd ..

muralikrishna@Shrita MINGW64 ~ (master)

$ git stash mehta

usage: git stash list [<options>]

or: git stash show [<stash>]

or: git stash drop [-q|--quiet] [<stash>]

or: git stash ( pop | apply ) [--index] [-q|--quiet] [<stash>]

or: git stash branch <branchname> [<stash>]

or: git stash [save [--patch] [-k|--[no-]keep-index] [-q|--quiet]

[-u|--include-untracked] [-a|--all] [<message>]]

or: git stash clear

muralikrishna@Shrita MINGW64 ~ (master)

$ git stash list

stash@{0}: WIP on newone: a25a947 Merge https://github.com/muralicloud/gogula

muralikrishna@Shrita MINGW64 ~ (master)

$ git stash drop mehta

'mehta' is not a stash reference

muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout mehta

M murali/gogula

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git checkout master

M murali/gogula

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~ (master)

$ git merge mehta

Updating a25a947..cadff03

Fast-forward

saai.html | 1 -

test/murali.txt | 1 +

test/newmurali.html | 1 +

3 files changed, 2 insertions(+), 1 deletion(-)

delete mode 100644 saai.html

create mode 100644 test/murali.txt

create mode 100644 test/newmurali.html

muralikrishna@Shrita MINGW64 ~ (master)

$ git log

commit cadff03d90694fa241cafa1ad791338ded12c2e0

Author: murali <murali.protek@gmail.com>

Date: Sun Apr 10 16:31:50 2016 -0700

dsglsngs

commit a25a9472aa1825a9303fc75d3a7ba275a95b9f74

Merge: 3f430cb bc8dbe5

Author: murali <murali.protek@gmail.com>

Date: Mon Mar 28 10:59:34 2016 -0700

Merge https://github.com/muralicloud/gogula

commit bc8dbe5686fa8f83e79d239500dc7645f204407e

Author: murali <murali.protek@gmail.com>

Date: Mon Mar 28 10:58:32 2016 -0700

Create fdfd

commit 3f430cbeccaecbb099b3826f77975b94ea922060

Author: murali <murali.protek@gmail.com>

Date: Mon Mar 28 10:42:24 2016 -0700

murali

commit 569c3693e097eac23068b2726d71e0799be8086b

Author: murali <murali.protek@gmail.com>

Date: Mon Mar 28 10:22:44 2016 -0700

first

commit 0518efb9c939fa2ea3e4e89390fc3919217a4023

Author: murali <murali.protek@gmail.com>

Date: Mon Mar 28 10:14:45 2016 -0700

Initial commit

commit 8dd1a42d92f58fc315a3087ab3302bcd8b7e7d0f

Author: murali <murali.protek@gmail.com>

Date: Fri Mar 25 14:27:56 2016 -0700

knjsdfkdsj

commit b8d024a4acac1c4b3f89bc8f38af282ccc75ca36

Author: murali <murali.protek@gmail.com>

Date: Fri Mar 25 13:57:58 2016 -0700

first commit

commit 8a75563812aee226e2618ee7ff6e3aa9687ce73a

Author: murali <murali.protek@gmail.com>

Date: Fri Mar 25 12:35:14 2016 -0700

muralikrishna@Shrita MINGW64 ~ (master)

$ git push

fatal: No configured push destination.

Either specify the URL from the command-line or configure a remote repository using

git remote add <name> <url>

and then push using the remote name

git push <name>

muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout mehta

M murali/gogula

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git push

fatal: No configured push destination.

Either specify the URL from the command-line or configure a remote repository using

git remote add <name> <url>

and then push using the remote name

git push <name>

muralikrishna@Shrita MINGW64 ~ (mehta)

$ ls

3D Objects/ IntelGraphicsProfiles/ newrepo/ protek1.html

AppData/ ITU-/ NOTICE README.md

Application Data@ ituday2/ nrepo/ Recent@

bin/ kuldip/ NTUSER.DAT RELEASE-NOTES

BullseyeCoverageError.txt lib/ ntuser.dat.LOG1 RemoteSystemsTempFiles/

class1/ libgit2/ ntuser.dat.LOG2 repo/

cloud3 LICENSE NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.0.regtrans-ms RUNNING.txt

Cloud4/ Links/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.1.regtrans-ms saiteja/

cloudrep/ Local Settings@ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.2.regtrans-ms saitejak/

conf/ logs/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.blf sample.txt

Contacts/ md NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TM.blf Saved Games/

Cookies@ Midterm/ NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000001.regtrans-ms Searches/

Creative Cloud Files/ murali/ NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000002.regtrans-ms SendTo@

Desktop/ murali.txt ntuser.ini Start Menu@

Documents/ muralione/ OneDrive/ temp/

Downloads/ Music/ Others/ Templates@

Favorites/ My Documents@ Pictures/ test/

fdfd my\_project/ practise/ Tracing/

file1.txt mycode/ practise.html Videos/

first.js/ myrepo/ PrintHood@ VirtualBox VMs/

git/ NetHood@ project/ webapps/

gitday2/ newdoc.txt/ protek/ work/

index.html newproject/ Protek1/ workspace/

muralikrishna@Shrita MINGW64 ~ (mehta)

$ cd test

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git push

fatal: No configured push destination.

Either specify the URL from the command-line or configure a remote repository using

git remote add <name> <url>

and then push using the remote name

git push <name>

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git add \*

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git commit -m "dfnff"

On branch mehta

Changes not staged for commit:

modified: ../murali/gogula (new commits)

Untracked files:

../.VirtualBox/

../.bash\_history

../.docker/

../.eclipse/

../.gitconfig

../.idlerc/

../.metadata/

../.oracle\_jre\_usage/

../.ssh/

../.vim/

../.viminfo

../3D Objects/

../AppData/

../BullseyeCoverageError.txt

../Cloud4/

../Contacts/

../Creative Cloud Files/

../Desktop/

../Documents/

../Downloads/

../Favorites/

../ITU-/

../IntelGraphicsProfiles/

../LICENSE

../Links/

../Midterm/

../Music/

../NOTICE

../NTUSER.DAT

../NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.0.regtrans-ms

../NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.1.regtrans-ms

../NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.2.regtrans-ms

../NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.blf

../NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TM.blf

../NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000001.regtrans-ms

../NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000002.regtrans-ms

../OneDrive/

../Others/

../Pictures/

../Protek1/

../RELEASE-NOTES

../RUNNING.txt

../RemoteSystemsTempFiles/

../Saved Games/

../Searches/

../Tracing/

../Videos/

../VirtualBox VMs/

../bin/

../cloud3

../cloudrep/

../conf/

../first.js/

../git/

../gitday2/

../index.html

../ituday2/

../lib/

../libgit2/

../md

../murali.txt

../muralione/

../my\_project/

../mycode/

../myrepo/

../newdoc.txt/

../newproject/

../newrepo/

../nrepo/

../ntuser.dat.LOG1

../ntuser.dat.LOG2

../ntuser.ini

../practise/

../project/

../repo/

../saitejak/

../sample.txt

../temp/

../webapps/

../workspace/

no changes added to commit

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git push

fatal: No configured push destination.

Either specify the URL from the command-line or configure a remote repository using

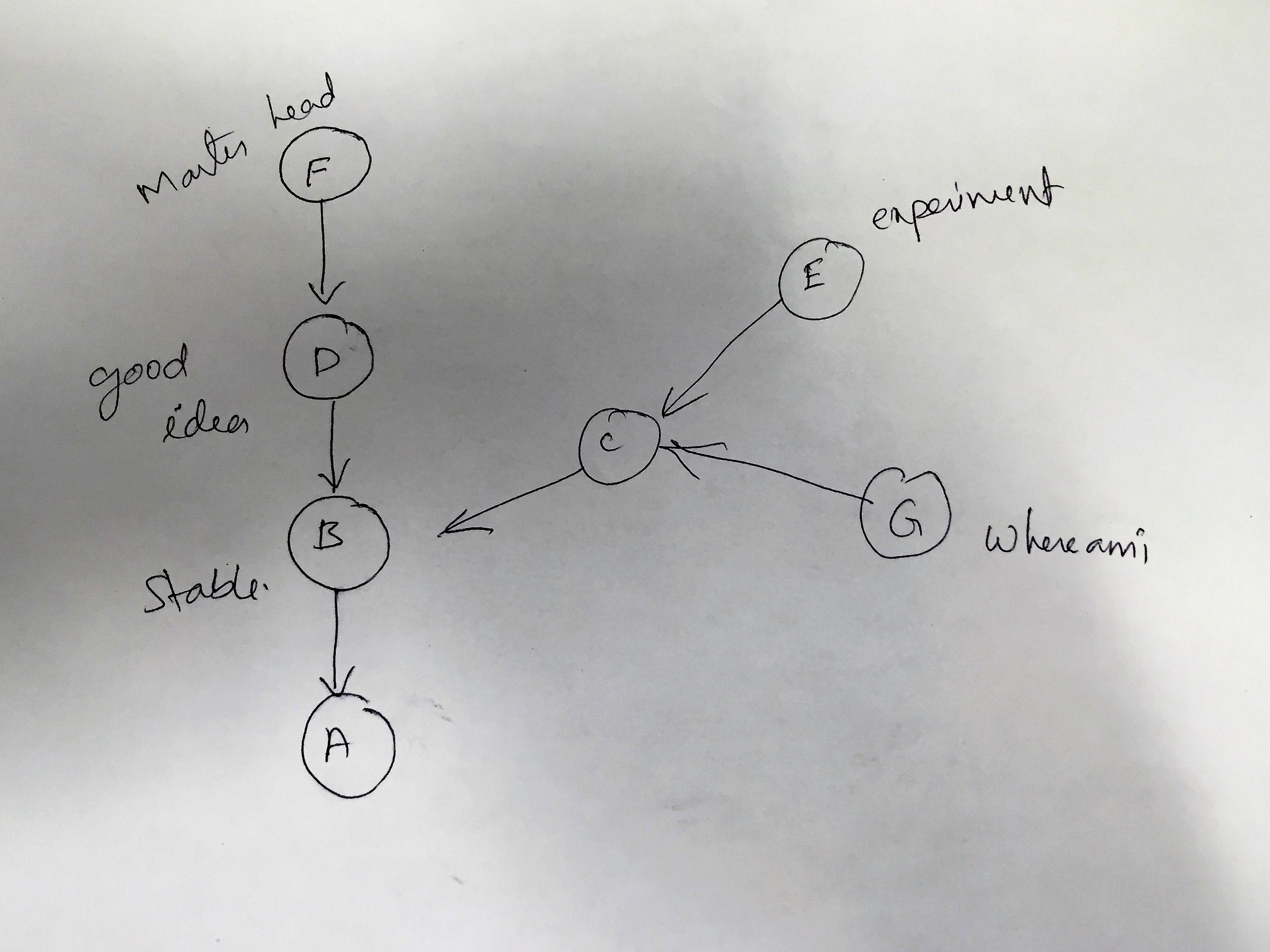
git remote add <name> <url>

and then push using the remote name

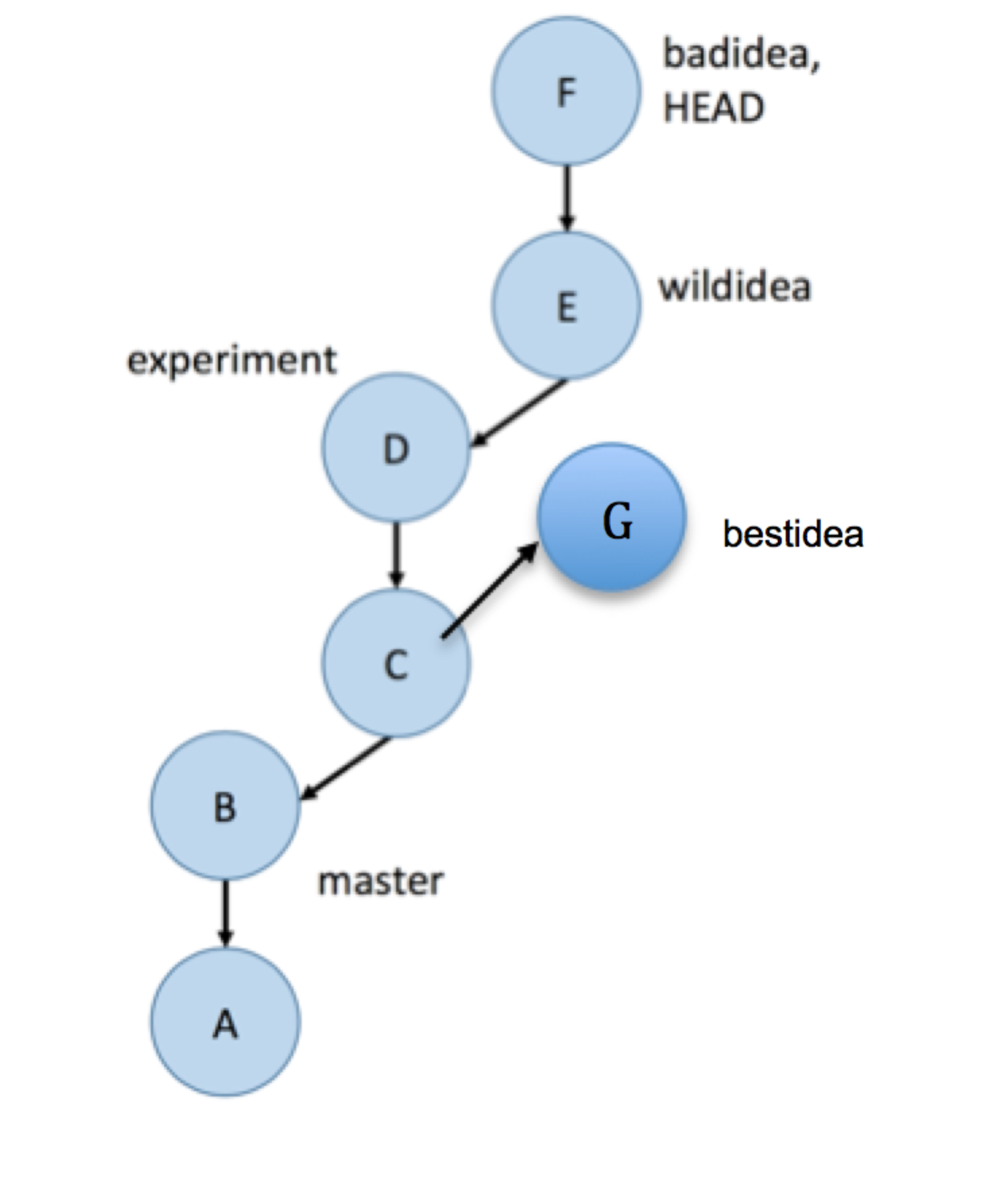
git push <name>

muralikrishna@Shrita MINGW64 ~/test (mehta)

1. Run the following commands on a given repo. Copy your git log below. Also upload your repo to remote repository on github and paste your link to the github repo. Also Draw a git Flow Diagram for following set of commands. You can draw it manually with hand and then upload the photo to EMS with Question number as file name.
2. git commit –m “A”
3. git commit –m “B”
4. git branch stable
5. git checkout –b experiment
6. git commit –m “C”
7. git checkout master
8. git commit –m “D”
9. gitbranchgoodidea
10. gitcheckout experiment
11. git branch whereami
12. git commit –m “E”
13. git checkout goodidea
14. git checkout master
15. git commit –m “F”
16. git checkout whereami
17. gitcommit–m“G”
18. git checkout master



1. Write a series of git commands (similar to what was given in the previous question) that would produce the following git graph. Make sure that branches and HEAD end up on the right commits by the end.



Answer)

* Git commit –m “A”
* Git commit –m “B”
* Git commit –m “C”
* Git branch “bestidea”
* Git commit –m “G”
* Git commit –m “H”
* Git branch “Experiment”
* Git checkout “Experiment”
* Git commit –m “D”
* Git checkout –b “Wildidea”
* Git commit –m “E”
* Git commit –m “F”
* Git Checkout –b “badidea”

5. A) After reading StackOverflow, Lucas decides that the correct way to make commits in git is to do git add .. Upon running git status, he discovers that as a result, he's staged the following files that he doesn't want to include in his next commit: .DS\_Store, npm-debug.log, passwords.txt, mydiary.txt. Advise Lucas (1) what to do to fix the immediate problem and also (2) what he could do specifically to prevent this from happening in the future (saying "don't trust everything you read on StackOverflow" is not sufficient).

1. My suggestion to Lucas is to use **git reset**

* Git reset .DS\_Store
* Git reset npm –debug.log
* Git reset passwords.txt
* Git reset mydairy.txt

B

i)After editing a source file for a few days, Ana decides that she likes the version she had a few days ago better than what she has now. She now wishes to replace the current version of the file with that previous version.

Ans: If Ana only editing one file, she can use **Git reset** to the commit she got.

ii)Ana doesn't remember the short hash of the most recent commit the previous version was last part of. Practically speaking, what would Ana have to do to find this out?

Ans: Ana can check **git log** on command line/terminal or check her commit on git.

iii)Once Ana finds this commit, how can she restore that previous version of the file? Note that she wants to restore just that file, not every file in the commit.

Ans: If ana finds the commit #, She can use the

**Git checkout <commit>/path/to/file**

6) Explain what is merge? Explain merge strategies? What is merge conflict? Explain how different ways of resolving merge conflict? Give an example of merge conflict please annotate and document your git command flow below. i.e. Create a branch/es, create a conflict and resolve it using git command. Show your work.

**Merge:**  Merge is joining two or more development histories together. In git it is like merging two or more branches or repositories or files.

**Merge Strategies:**

The merge mechanism (git merge and git pull commands) allows the backend merge strategiesto be chosen with -s option. Some strategies can also take their own options, which can be passed by giving -X<option> arguments to git merge and/or git pull.

**resolve**

This can only resolve two heads (i.e. the current branch and another branch you pulled from) using a 3-way merge algorithm. It tries to carefully detect criss-cross merge ambiguities and is considered generally safe and fast.

**recursive**

This can only resolve two heads using a 3-way merge algorithm. When there is more than one common ancestor that can be used for 3-way merge, it creates a merged tree of the common ancestors and uses that as the reference tree for the 3-way merge. This has been reported to result in fewer merge conflicts without causing mismerges by tests done on actual merge commits taken from Linux 2.6 kernel development history. Additionally this can detect and handle merges involving renames. This is the default merge strategy when pulling or merging one branch.

The recursive strategy can take the following options:

**ours**

This option forces conflicting hunks to be auto-resolved cleanly by favoring our version. Changes from the other tree that do not conflict with our side are reflected to the merge result. For a binary file, the entire contents are taken from our side.

This should not be confused with the ours merge strategy, which does not even look at what the other tree contains at all. It discards everything the other tree did, declaring our history contains all that happened in it.

**theirs**

This is the opposite of ours.

**patience**

With this option, merge-recursive spends a little extra time to avoid mismerges that sometimes occur due to unimportant matching lines (e.g., braces from distinct functions). Use this when the branches to be merged have diverged wildly. See also [git-diff[1]](https://git-scm.com/docs/git-diff) --patience.

**diff-algorithm=[patience|minimal|histogram|myers]**

Tells merge-recursive to use a different diff algorithm, which can help avoid mismerges that occur due to unimportant matching lines (such as braces from distinct functions). See also [git-diff[1]](https://git-scm.com/docs/git-diff) --diff-algorithm.

**ignore-space-change**

**ignore-all-space**

**ignore-space-at-eol**

Treats lines with the indicated type of whitespace change as unchanged for the sake of a three-way merge. Whitespace changes mixed with other changes to a line are not ignored. See also [git-diff[1]](https://git-scm.com/docs/git-diff) -b, -w, and --ignore-space-at-eol.

* If their version only introduces whitespace changes to a line, our version is used;
* If our version introduces whitespace changes but their version includes a substantial change, theirversion is used;
* Otherwise, the merge proceeds in the usual way.

**renormalize**

This runs a virtual check-out and check-in of all three stages of a file when resolving a three-way merge. This option is meant to be used when merging branches with different clean filters or end-of-line normalization rules. See "Merging branches with differing checkin/checkout attributes" in[gitattributes[5]](https://git-scm.com/docs/gitattributes) for details.

**no-renormalize**

Disables the renormalize option. This overrides the merge.renormalize configuration variable.

**no-renames**

Turn off rename detection. See also [git-diff[1]](https://git-scm.com/docs/git-diff) --no-renames.

**find-renames[=<n>]**

Turn on rename detection, optionally setting the similarity threshold. This is the default. See also [git-diff[1]](https://git-scm.com/docs/git-diff) --find-renames.

**rename-threshold=<n>**

Deprecated synonym for find-renames=<n>.

**subtree[=<path>]**

This option is a more advanced form of subtree strategy, where the strategy makes a guess on how two trees must be shifted to match with each other when merging. Instead, the specified path is prefixed (or stripped from the beginning) to make the shape of two trees to match.

**octopus**

This resolves cases with more than two heads, but refuses to do a complex merge that needs manual resolution. It is primarily meant to be used for bundling topic branch heads together. This is the default merge strategy when pulling or merging more than one branch.

**ours**

This resolves any number of heads, but the resulting tree of the merge is always that of the current branch head, effectively ignoring all changes from all other branches. It is meant to be used to supersede old development history of side branches. Note that this is different from the -Xours option to therecursive merge strategy.

**subtree**

This is a modified recursive strategy. When merging trees A and B, if B corresponds to a subtree of A, B is first adjusted to match the tree structure of A, instead of reading the trees at the same level. This adjustment is also done to the common ancestor tree.

With the strategies that use 3-way merge (including the default, recursive), if a change is made on both branches, but later reverted on one of the branches, that change will be present in the merged result; some people find this behavior confusing. It occurs because only the heads and the merge base are considered when performing a merge, not the individual commits. The merge algorithm therefore considers the reverted change as no change at all, and substitutes the changed version instead.

**Merge Conflict:**

During a merge, the working tree files are updated to reflect the result of the merge. Among the changes made to the common ancestor’s version, non-overlapping ones (that is, you changed an area of the file while the other side left that area intact, or vice versa) are incorporated in the final result verbatim. When both sides made changes to the same area, however, Git cannot randomly pick one side over the other, and asks you to resolve it by leaving what both sides did to that area.

**Example in my Previous task**

**EX 1)**

muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout -b mehta

M murali/gogula

D saai.html

Switched to a new branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

$ ^C

muralikrishna@Shrita MINGW64 ~ (mehta)

$ vi murali.txt

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git add \*

fatal: Not a git repository: AppData/Local/GitHub/IgnoreTemplates\_a66c3719071da6d865a984bb8d6bfb5bcd775ec8/../.git/modules/gitignore

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git commit

Aborting commit due to empty commit message.

muralikrishna@Shrita MINGW64 ~ (mehta)

$ mkdir test

muralikrishna@Shrita MINGW64 ~ (mehta)

$ cd test

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ vi murali.txt

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git add \*

warning: LF will be replaced by CRLF in test/murali.txt.

The file will have its original line endings in your working directory.

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

D saai.html

A test/murali.txt

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test (master)

$ ls

murali.txt

muralikrishna@Shrita MINGW64 ~/test (master)

$ ^C

muralikrishna@Shrita MINGW64 ~/test (master)

$ git checkout mehta

M murali/gogula

D saai.html

A test/murali.txt

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ ls

murali.txt

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ vi newmurali.html

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git add \*

warning: LF will be replaced by CRLF in test/newmurali.html.

The file will have its original line endings in your working directory.

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

D saai.html

A test/murali.txt

A test/newmurali.html

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test (master)

$ ls

murali.txt newmurali.html

muralikrishna@Shrita MINGW64 ~/test (master)

$ git checkout mehta

M murali/gogula

D saai.html

A test/murali.txt

A test/newmurali.html

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git commit -m "dsglsngs"

[mehta warning: LF will be replaced by CRLF in test/murali.txt.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in test/newmurali.html.

The file will have its original line endings in your working directory.

cadff03] dsglsngs

warning: LF will be replaced by CRLF in test/murali.txt.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in test/newmurali.html.

The file will have its original line endings in your working directory.

3 files changed, 2 insertions(+), 1 deletion(-)

delete mode 100644 saai.html

create mode 100644 test/murali.txt

create mode 100644 test/newmurali.html

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test

$ ls

ls: cannot open directory .: Permission denied

muralikrishna@Shrita MINGW64 ~/test

$ git checkout master

fatal: Unable to read current working directory: No error

muralikrishna@Shrita MINGW64 ~/test

$ git checkout mehta

fatal: Unable to read current working directory: No error

muralikrishna@Shrita MINGW64 ~/test

$ cd ..

muralikrishna@Shrita MINGW64 ~ (master)

$ cd test

bash: cd: test: No such file or directory

muralikrishna@Shrita MINGW64 ~ (master)

$ git branch

checkout

chekout

masster

\* master

mehta

murali

newone

muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout mehta

M murali/gogula

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

$ ls

3D Objects/ IntelGraphicsProfiles/ newrepo/ protek1.html

AppData/ ITU-/ NOTICE README.md

Application Data@ ituday2/ nrepo/ Recent@

bin/ kuldip/ NTUSER.DAT RELEASE-NOTES

BullseyeCoverageError.txt lib/ ntuser.dat.LOG1 RemoteSystemsTempFiles/

class1/ libgit2/ ntuser.dat.LOG2 repo/

cloud3 LICENSE NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.0.regtrans-ms RUNNING.txt

Cloud4/ Links/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.1.regtrans-ms saiteja/

cloudrep/ Local Settings@ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.2.regtrans-ms saitejak/

conf/ logs/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.blf sample.txt

Contacts/ md NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TM.blf Saved Games/

Cookies@ Midterm/ NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000001.regtrans-ms Searches/

Creative Cloud Files/ murali/ NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000002.regtrans-ms SendTo@

Desktop/ murali.txt ntuser.ini Start Menu@

Documents/ muralione/ OneDrive/ temp/

Downloads/ Music/ Others/ Templates@

Favorites/ My Documents@ Pictures/ test/

fdfd my\_project/ practise/ Tracing/

file1.txt mycode/ practise.html Videos/

first.js/ myrepo/ PrintHood@ VirtualBox VMs/

git/ NetHood@ project/ webapps/

gitday2/ newdoc.txt/ protek/ work/

index.html newproject/ Protek1/ workspace/

muralikrishna@Shrita MINGW64 ~ (mehta)

$ cd test

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ ls

murali.txt newmurali.html

muralikrishna@Shrita MINGW64 ~/test (mehta)

$ git checkout master

M murali/gogula

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~/test

$ cd ..

muralikrishna@Shrita MINGW64 ~ (master)

$ git stash mehta

usage: git stash list [<options>]

or: git stash show [<stash>]

or: git stash drop [-q|--quiet] [<stash>]

or: git stash ( pop | apply ) [--index] [-q|--quiet] [<stash>]

or: git stash branch <branchname> [<stash>]

or: git stash [save [--patch] [-k|--[no-]keep-index] [-q|--quiet]

[-u|--include-untracked] [-a|--all] [<message>]]

or: git stash clear

muralikrishna@Shrita MINGW64 ~ (master)

$ git stash list

stash@{0}: WIP on newone: a25a947 Merge https://github.com/muralicloud/gogula

muralikrishna@Shrita MINGW64 ~ (master)

$ git stash drop mehta

'mehta' is not a stash reference

muralikrishna@Shrita MINGW64 ~ (master)

$ git checkout mehta

M murali/gogula

Switched to branch 'mehta'

muralikrishna@Shrita MINGW64 ~ (mehta)

$ git checkout master

M murali/gogula

Switched to branch 'master'

muralikrishna@Shrita MINGW64 ~ (master)

$ git merge mehta

Updating a25a947..cadff03

Fast-forward

saai.html | 1 -

test/murali.txt | 1 +

test/newmurali.html | 1 +

3 files changed, 2 insertions(+), 1 deletion(-)

delete mode 100644 saai.html

create mode 100644 test/murali.txt

create mode 100644 test/newmurali.html

**Ex 2)**

muralikrishna@Shrita MINGW64 ~/Others (master)

$ git add \*

warning: LF will be replaced by CRLF in test.html.

The file will have its original line endings in your working directory.

muralikrishna@Shrita MINGW64 ~/Others (master)

$ git commit -m "new file"

[master 8499232] new file

warning: LF will be replaced by CRLF in test.html.

The file will have its original line endings in your working directory.

1 file changed, 1 insertion(+)

create mode 100644 test.html

muralikrishna@Shrita MINGW64 ~/Others (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 directory clean

muralikrishna@Shrita MINGW64 ~/Others (master)

$ git push

warning: push.default is unset; its implicit value has changed in

Git 2.0 from 'matching' to 'simple'. To squelch this message

and maintain the traditional behavior, use:

git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

git config --global push.default simple

When push.default is set to 'matching', git will push local branches

to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'

behavior, which only pushes the current branch to the corresponding

remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.

(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode

'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 3, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 288 bytes | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/muralicloud/Others.git

e7c2a27..8499232 master -> master

muralikrishna@Shrita MINGW64 ~/Others (master)

$ git status

On branch master

Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean

muralikrishna@Shrita MINGW64 ~/Others (master)

$ git pull

Already up-to-date.

muralikrishna@Shrita MINGW64 ~/Others (master)

$ cd ..

muralikrishna@Shrita MINGW64 ~ (master)

$ ls

3D Objects/ index.html NOTICE protek1.html

AppData/ IntelGraphicsProfiles/ nrepo/ README.md

Application Data@ ITU-/ NTUSER.DAT Recent@

bin/ ituday2/ ntuser.dat.LOG1 RELEASE-NOTES

BullseyeCoverageError.txt kuldip/ ntuser.dat.LOG2 RemoteSystemsTempFiles/

class1/ lib/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.0.regtrans-ms RUNNING.txt

cloud3 libgit2/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.1.regtrans-ms saiteja/

Cloud4/ LICENSE NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.2.regtrans-ms saitejak/

cloudrep/ Links/ NTUSER.DAT{174c13b1-a57d-11e5-9203-d7fe1cc80fb7}.TxR.blf sample.txt

conf/ Local Settings@ NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TM.blf Saved Games/

Contacts/ logs/ NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000001.regtrans-ms Searches/

Cookies@ md NTUSER.DAT{174c13b2-a57d-11e5-9203-d7fe1cc80fb7}.TMContainer00000000000000000002.regtrans-ms SendTo@

Creative Cloud Files/ Midterm/ ntuser.ini Start Menu@

Desktop/ murali/ OneDrive/ temp/

Documents/ muralione/ Others/ Templates@

Downloads/ Music/ Pictures/ Tracing/

Favorites/ My Documents@ practise/ Videos/

fdfd my\_project/ practise.html VirtualBox VMs/

file1.txt mycode/ PrintHood@ webapps/

first.js/ myrepo/ project/ work/

git/ NetHood@ protek/ workspace/

gitday2/ newproject/ Protek1/

muralikrishna@Shrita MINGW64 ~ (master)

$ cd Midterm

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ ls

midterm.html README.md

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ git merge Others

merge: Others - not something we can merge

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ git merge "Others"

merge: Others - not something we can merge

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ git init

Reinitialized existing Git repository in C:/Users/muralikrishna/Midterm/.git/

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ git pull https://github.com/muralicloud/Others.git

warning: no common commits

remote: Counting objects: 6, done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0

Unpacking objects: 100% (6/6), done.

From https://github.com/muralicloud/Others

\* branch HEAD -> FETCH\_HEAD

Auto-merging README.md

CONFLICT (add/add): Merge conflict in README.md

Automatic merge failed; fix conflicts and then commit the result.

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ ls

midterm.html README.md test.html

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ git fetch

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ ls

midterm.html README.md test.html

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ git diff

**diff --cc README.md**

**index fe8790d,7d789cc..0000000**

**--- a/README.md**

**+++ b/README.md**

@@@ -1,2 -1,1 +1,6 @@@

-# Others

++<<<<<<< HEAD

+# Midterm

+For Exam Purpose

++=======

++# Others

++>>>>>>> 8499232a4d03c045de4272f18391c6e37e610431

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ git fetch origin

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ vi newone.txt

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ git add \*

warning: LF will be replaced by CRLF in newone.txt.

The file will have its original line endings in your working directory.

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ git status

On branch master

Your branch is up-to-date with 'origin/master'.

All conflicts fixed but you are still merging.

(use "git commit" to conclude merge)

Changes to be committed:

modified: README.md

new file: newone.txt

new file: test.html

muralikrishna@Shrita MINGW64 ~/Midterm (master|MERGING)

$ git stash save "newone.txt"

Saved working directory and index state On master: newone.txt

HEAD is now at fcbdef1 fgjg

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ git status

On branch master

Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ git stash

No local changes to save

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ git stash list

stash@{0}: On master: newone.txt

muralikrishna@Shrita MINGW64 ~/Midterm (master)

$ git stash pop

On branch master

Your branch is up-to-date with 'origin/master'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: newone.txt

new file: test.html

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: README.md