

# GIT SPARSE CHECKOUT : SPARSE INDEX FEATURE

## INTEGRATION

### Personal Information

- Name: Debra Obondo
- Email: [debraobondo@gmail.com](mailto:debraobondo@gmail.com)
- University: Jomo Kenyatta University of Agriculture and Technology
- Telephone Number: +254743635117
- Course: B.Sc Electronic and Computer Engineering
- Github: <https://github.com/fobiasic07>
- LinkedIn: <https://www.linkedin.com/in/debra-obondo>
- Timezone: Nairobi (GMT+3)
- Location: Nairobi, Kenya

## Abstract

Currently, the “git sparse-checkout” feature has a few commands integrated within it which are likely to be phased out in future as a result of their different functionality within this mode as compared to the normal mode, such as the “git commit -a” command. We appreciate that it helps developers working in huge repositories by picking a subset of specific directories and files from the current working tree for ease of edit and access. It uses the “git checkout branch” with some aspect of “.gitignore” to do this.

My contribution toward this project is to help Git achieve its purpose of integrating more commands with the sparse index feature and to ensure their effectiveness and speed in this mode for convenient use by other developers working within git. My aim is to help this goal be achieved faster and properly during this GSoC period.

The integration of these commands will help git open source software take its next development step as the best in “distributed version systems” controller. This will also help to take pressure off contributors especially in large repositories by allowing them to work with a small set of files and to shift to the sparse-checkout mode without incident.

## Deliverables

The aim is to make git sparse-checkout more convenient to use for developers, especially when large repositories are involved, by accelerating git commands in sparse-checkout cone mode. I shall do this by:

- Doing weekly tests on the various commands listed in my timeline.

- Writing weekly notes in the README.md for the commands worked, tests passed, changes made and best functionality.
- Giving comprehensible fortnightly reports on all the details of the commands for future use by other developers.
- Final report on the entire project, challenges faced, solutions established and future recommendations on similar projects.

## Project Implementation

Majority of the project shall be implemented by use of Bash scripting and C.

## Programming Plan

I have divided my programming plan into 4 parts, which would be as follows:

PART 1(Pre- Program): This is before the official start of the program. I plan to familiarize myself with the cone mode and sparse checkout, understand its functionality, test already existing commands and do micro projects in my github account on the same and test skills learnt on the <https://github.com/git/git/blob/master/t/t1092-sparse-checkout-compatibility.sh> file.

PART 2 (Phase 1): This is during the first half of the program, I have organized my timeline to achieve completion of about half of the total project's work, including tests and reports of work done.

PART 3 (Phase 2): During the final half of the program, I plan to have completed at least 90% of the general project's commands, together with reports on their applicability.

PART 4 (Post-Program):

After the effort and work put into this project, it is only justified to continue contributing toward the growth of the checkout-sparse mode. I, therefore, plan to keep looking into further commands that should be added to this mode and continuously share all required information and documentation of the same. I shall also look into mentoring the next generation of developers to take over this project and to help their work easen.

## Timeline

Now - 20th May

- Understand organization and structure better.
- Familiarize and Understand git sparse-checkout and Index Feature further.
- Complete compulsory and few optional microprojects.

- Bonding with mentors and community members.
- Gaining understanding of Git commands yet to be explored.
- Read through various git commands documentation and practice on learnt concepts.

#### 21st May - 27th May (Week 1)

- Community bonding and Interaction with Mentor(s) assigned.
- Begin working on git mv command in sparse checkout.
- Test the command and write short, graspable notes on functionality
- Discussion with Mentor on weekly work, challenges and concepts.

#### 28th May - 3rd June (Week 2)

- Test git rm command on sparse-checkout and observe excluded paths.
- Check git grep functionality and edit accordingly
- Weekly discussion with mentor on the same.
- Submission of fortnightly report on commands worked, their functionality and use for future developers.

#### 4th June - 10th June (Week 3)

- Working on the git rev-parse command.
- Short report on the workability and any changes made.
- Weekly discussion with mentor on challenges faced and best solutions of the command.

#### 11th June - 17th June (Week 4)

- Assess git fsck command in sparse-checkout cone mode.
- Short details on effectiveness and changes made, if any.
- Mentor remote meet-up.
- Fortnightly report on commands worked and their efficiency.

#### 18th June - 24th June (Week 5)

- Tests of git check-attr command and best functionality.
- Weekly discussion with mentor on the worked command.

#### 25th June - 1st July (Week 6)

- git describe tests and account of any changes done.
- Mentor-mentee discussion.
- Fortnightly report on commands worked through the two weeks.

#### 2nd July - 8th July (Week 7)

- Look through the git diff command documentation.
- git diff-files command testing, editing and short-note write-up.
- Weekly meeting with mentor.

9th July - 15th July (Week 8)

- Complete any git diff-files test yet to be done.
- Begin on git diff-index command tests and minimize on it.
- Weekly discussion with mentor.
- Fortnightly report submission.

16th July - 22nd July (Week 9)

- Completion of git diff-index command.
- git diff-tree tests and report.
- Weekly mentor-mentee meet-up.

23rd July - 29th July (Week 10)

- Go through git tree commands documentation.
- Tests on git worktree command with effective implementation.
- Weekly discussion with mentor.
- First-half project report submission and evaluation

★ First Project Evaluation

30th July - 5th August (Week 11)

- Completion of any remainder work from the first half phase.
- Tests on git worktree command.
- Commencement of git write-tree command.
- Weekly mentor-mentee discussion.

6th Aug - 12th August (Week 12)

- Completion of git write-tree tests and report.
- Weekly discussion with mentor.
- Fortnightly report on commands completed.

13th August - 19th August (Week 13)

- git apply command tests and report.
- Weekly Mentor meet-up.

20th August - 25th August (Week 14)

- Completion of git apply tests and report
- Start of git am command test.
- Mentor-mentee weekly discussion.
- Fortnightly reports on commands completed, challenges faced and solution paths forked.

26th August - 2nd September (Week 15)

- Completion of git am command tests.
- git checkout--worker tests and report.

- Weekly discussion with mentor.

3rd September - 9th September(Week 16)

- Working on the git merge-index command.
- Begin working on the git rerere command.
- Mentor-mentee discussion on project progress and any remaining work.

9th September - 12th September (Week 17)

- git rerere command completion.
- Submission of Project Final Report.

★ Final Project Evaluation

## Perceived Obstacles and Challenges

- Overflow of work into the following week.

In such a case scenario, I will ensure to discuss with my mentor in advance so as to find a quick and convenient solution, such as increasing hours spent contributing.

- Test fails

For this occurrence, together with my mentor, we shall explore other solutions to solve the problem within the test.

## About Me

Hello, I am a fourth year B.Sc student taking Electronics and Computer Engineering at Jomo Kenyatta University of Agriculture and Technology, Kenya. I am also a member of the ALX 2022 Programme.

For a long time, I didn't have a specific career path I had wanted to pursue until I had a few tries with software, which I had initially considered complicated, but now I am fully interested and investing in it. I was referred to GSoC by a friend about 2 weeks ago resulting in my delayed application and submission.

I have experience with C from my course and ALX practice and have recently gained skills and knowledge in bash scripting and git, which I would like to improve and grow in. I believe contributing to this open source project will help me to further grow my skills and help others who have had a similar mentality to my previous one.

Equipped with this chance, I shall also use the skills learnt to help future developers on similar projects and to continue contributing to git and other open-source projects.

## Requirements

- This is the micro project required while submitting the proposal for the GSoC 2022 program: <https://github.com/fobiasic07/git-gsoc-2022-microproject>.

By the submission time of this proposal, the microproject is yet to be done due to the delay in application and the time limit to deadline, thus, project discussion is absent, but shall be included in the README.md of the above forked repository within a week from GSoC submission deadline.

## Obligations and Schedule Conflicts

I am currently a software student learning through ALX, which is quite similar to the Holberton Software Programme, which shall be going on during this period. It is through this course, however, that I have gained interest in working with Git for the growth of distributed version control systems.

During the GSoC period, I shall have begun my first semester of my fourth year. I shall be able to contribute 4-5 hours a day(25 hours a week) toward the project. Between 8th and 29th of August, I shall be sitting for my End of Semester Examinations and will be giving about 2-3 hours a day(15 hours a week). After the Examinations, I shall be on a fortnight break and will contribute 6-7 hours a day(35 hours a week).

Given the opportunity, I will give this project my best and ensure to allocate my time adequately to meet the deadlines in good time. In the case of any mishap, I shall be sure to communicate with the mentors in good time for efficient progress and completion of this project.

## References

- Git SoC 2022 Ideas : <https://git.github.io/SoC-2022-Ideas/>
- Git sparse-checkout documentation : <https://git-scm.com/docs/git-sparse-checkout>
- Git Soc 2022 Applicant Microprojects : <https://git.github.io/SoC-2022-Microprojects/>
- Making a mono-repo feel small blog : <https://github.blog/2021-11-10-make-your-monorepo-feel-small-with-gits-sparse-index/>