

Google Summer of Code

BY MARIUSZ ZABORSKI

Google Summer of Code (GSoC) 2020 ended in September. This amazing initiative allows young students to gain a lot of experience while contributing to open-source organizations. FreeBSD project have participated in the GSoC each year from the start. In this article, we introduce this project from the student side and the mentor side.

Google Summer of Code (GSoC) is an annual, international program that Google started in 2005. The goal is to expose students to an open-source project, and the program is designed as a vacation scholarship for students. Instead of looking for internships or going on vacation, students can spend their summer coding for an open-source organization.

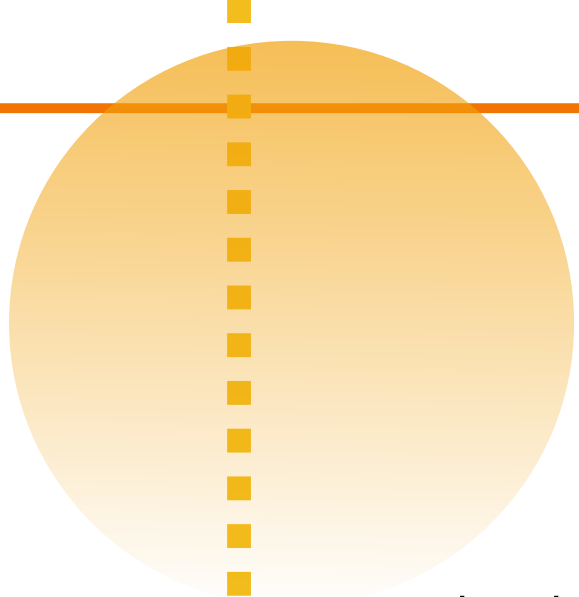
Students are obligated to work almost full time on the project of their choice, and Google funds a stipend for them. The amount varies each year. Currently, the scholarship amount depends on the student's location and it ranges from \$3,000 to \$7,000 USD.

Fortunately, the students are not alone, as GSoC requires students to work with a mentor. The mentors are members of, or contributors to, an open-source organization that is interested in the topic of the student's work.

GSoC starts way before summer. Google opens applications in January of each year and during this time, open-source projects can apply to join the program. A good practice is to prepare even earlier and constantly collect ideas for GSoC projects. Create a simple wiki page or Google Doc and collect potential projects for students to work on. The projects can vary from something simple to something very complicated, but you have to keep in mind that students should be able to provide some kind of code during the summer. It is very important to offer interesting projects; otherwise, students may choose a different open-source organization. Also, remember that a person from the organization (the mentor) should be responsible for the project. This person has to be able to help the student on their journey.

After a couple months, the open-source organizations are granted or denied acceptance in the program—they usually received word at the end of February. After that, students are given some time to look for open-source organizations and projects they would be interested in. They can discuss the ideas through mailing lists, IRC, Slack, or anywhere else. If they aren't interested in any of the projects listed, students can suggest a project themselves and then an open-source organization can look for a mentor for the project.

In March, the open-source community reviews the students' applications and assigns mentors. GSoC organizers inform each open-source project of how many seats they have. Unfortunately, there won't be enough seats for all students. The applications will be reviewed based



on the quality of the research done by the student, their knowledge of the subject, and their level of engagement in the last few months. Some good advice for students is to show interest as early as you can, think about the challenges of the project, do some research, discuss the challenges with the community, and even look into the code a little. Nobody expects you to know the code like the

back of your hand—your mentor will probably help you with the source code—but going the extra mile to look into things yourself demonstrates your engagement.

Mentors should not only have expertise in the project's subject matter but should also be integrated with the open-source community. If there is a problem they can't help with, or the student needs additional access to something, the mentor should know where to go. They should know the project code guide and be able to commit the code or get someone to commit the code. For some students, getting their first commit to open source may be more important than a scholarship (and you want to focus on those). Some projects may have more than one mentor.

After students are accepted, the community bonding and coding starts. This is the time for students' hard work. At FreeBSD, we have a guide for what students should do during this period. One of the things is to maintain a wiki page of their project. Students should try to update it as often as they can, which helps with project visibility in the organization. It is good to document the issues and challenges that we encounter during the summer because many projects won't be finished by the end of the program. By documenting the work that was done, we can pass it on to somebody else after GSoC.

Students should also send weekly status updates to the mailing list. This activity motivates students to share and document what they have accomplished and helps with the project's visibility. It also helps with community bonding. There may be some developers in the group who can help with an issue or have a different view of the project. It is really worth sharing status updates.

Last but not least, students should push their code as often as possible. The code quality doesn't really matter—you can always improve it—but it is much, much easier to discuss something when you see the code. Some students send code only once or twice a month. That doesn't work. After a month of coding, we may find that we should go in a totally different direction, and it is hard for mentors to help students when they don't see the code. Mentors should encourage students to push their code often and not be afraid to show work in progress.

Most of the time, the project is done online. Only a few students have the chance to work with their mentors on-site every day. Communication may be hard sometimes, especially if the mentor and students are in different time zones. Emails between students and mentors may seem too formal and students may be reluctant to write emails. Many students may also treat the relationship like a professor-student relationship. If you never see your mentor in person, it can be hard to imagine how they will react. Students and mentors should try to reduce the communication overhead. Communicating over an IRC, Slack, or Discord is much better for this purpose than email.

Mentors should also remember that the project is theirs, and they are responsible for it. If you don't respond to questions for weeks, then the student may be discouraged from working on the project. You have to be there for the students and help them, but remember that the real work should be done by the students. You're just there to help manage the process and help with communication with the community that you know so well.

Mentors evaluate their students a few times over the summer. (In previous years, there were two evaluations. Now, there are three.) The mentors report the students' progress to Google. Remember, if a mentor and student don't submit the evaluation on time, Google will cancel the project. Google pays the students after each successful evaluation.

If the project is going well, then great and it's not complicated. The mentor simply passes the student on to the next stage. If the student hasn't written a single line of code, then the next part is very unpleasant but still simple.

Unfortunately, there is also a middle ground. Some students are very active at the beginning of the project and then suddenly disappear. Some students rush and do all the work just before the evolution. Sometimes, it is hard to know how much work the student has done because of the remote working situation (this is why code pushing is so important). The most important thing is for mentors to ask themselves if they still want to work with the student. If not, they have to make a hard decision.

If there is still hope, mentors must tell their students what they should improve—and if they don't perform better, then they will fail them on the next evaluation. If you are doing more work than your student, then you have to fail the student. If you constantly struggle to engage your students, then you have to fail them. Student engagement is much more important than the quality of the code they produce.

There is no better way for open-source organizations to obtain fresh blood. Most students are not exposed to niche open-source projects. GSoC allows students to get to know a project from the inside out. They may contribute to the project (if they do, that's perfect), but it's not the most important part. The most important thing is to show them a great open-source environment and allow them to learn from it. If we do that, more and more students will stay in this environment, and everyone will gain from it. The students will gain knowledge that would otherwise be hard to get and earn nice CV entries that may open some new doors for them.

History shows that a lot of students who started a GSoC project continued to contribute to the project many years later. In this author's opinion, that should be the goal of any open-source organization involved in GSoC. We have to remember that the students' code won't be ideal, but if they stay on the project for longer than those three months, the gains for the organization will be much more significant than just this single project. Many students go on to become GSoC mentors the next year!

We hope to see you as a student or mentor at the next GSoC!

MARIUSZ ZABORSKI is a QA and Dev manager at Fudo Security. He completed a successful Google Summer of Code project in 2013 and mentored a few GSoC students for the FreeBSD Project. Mariusz has been the proud owner of his FreeBSD commit bit since 2015. His main areas of interest are OS security and low-level programming. At Fudo Security, Mariusz leads a team developing the most advanced solutions for monitoring, recording, and controlling traffic in IT infrastructure. In 2018, Mariusz organized the Polish BSD User Group. In his free time, he enjoys blogging at <https://oshogbo.vexillum.org>.