

# Project 1: User Programs

Welcome to Project User Programs! After completing Project Pregame, you are probably left with a lot of questions on how Pintos works in general and the validity of the “fix” you implemented. Don’t worry. This project will have you implement some core functionalities such as argument passing as well as essential syscalls and floating point operations that will fill in some holes from Project Pregame.

The details of this assignment can be found in the [Tasks](#) section. However, you may find it helpful to first read through some of the [CS 162 Pintos documentation](#). This will likely help you understand the required tasks.

Even if you read through some of these during Project 0, we recommend (re)reading through the following sections of the [Pintos documentation](#):

- [User Programs](#)
- [Floating Point](#)
- [Development](#)

---

## Setup

First, log into your Workspace and check if your `group` folder is initialized correctly

```
cd ~/code/group  
git remote -v
```

If your folder was set up correctly, you should see the following output:

```
staff https://github.com/Berkeley-CS162/group0.git (fetch)  
staff https://github.com/Berkeley-CS162/group0.git (push)
```

If your output doesn’t match the above, reinitialize your `group` folder.

```
rm -rf ~/code/group
git clone -o staff https://github.com/Berkeley-CS162/group0.git ~/code/group
cd ~/code/group
```

This should allow you to pull the starter code with `git pull staff main`.

Next, add a pre-commit hook that will format your code according to our linting standards.

```
chmod +x ~/code/group/.pre-commit.sh
ln -s -f ~/code/group/.pre-commit.sh ~/code/group/.git/hooks/pre-commit
```

This will ensure that before you make a commit, your code will be nicely formatted. Part of your project grade will be determined on your code quality, so it is strongly recommended that you install this formatter. See [Code](#) for more information.

Finally, head over to the [group dashboard on the autograder](#). If you are not registered as a group, you should do that first. One person should create a group and invite the other members of your group. Each member should then visit the group dashboard to accept the invitation.

Once your group is created, click on the GitHub logo from the group dashboard. This will take you to your group's repo on GitHub. If you are unable to access the repo (eg. if you see a 404 error), then make sure to check your email for an invitation. The invite may have expired, in which case you should post in an appropriate Ed thread or make a private post. A staff member will resend the invitation. Please make sure to accept it promptly!

Once you find your repo page on GitHub, copy the SSH URI, which should look like

```
git@github.com:Berkeley-CS162/groupX
```

where `x` is your group number. Head back over to your Workspace and add this as a remote, making sure to replace the `x`:

```
> git remote add group git@github.com:Berkeley-CS162/groupX.git
```

To double check everything was properly set up, run the following commands:

```
> git remote -v
group git@github.com:Berkeley-CS162/groupX.git (fetch)
```

```
group git@github.com:Berkeley-CS162/groupX.git (push)
staff https://github.com/Berkeley-CS162/group0.git (fetch)
staff https://github.com/Berkeley-CS162/group0.git (push)
> git remote show group
* remote group

Fetch URL: git@github.com:Berkeley-CS162/groupX.git
Push URL: git@github.com:Berkeley-CS162/groupX.git
HEAD branch: (unknown)
```