# Team Project
## 2021 Spring, SWPP

## 1. Goal

In this project, you'll work as a team with other students to write a simple compiler that reads an LLVM IR program and emits an optimized assembly for an imaginary machine.

Each team should

- Use LLVM **12.0** for your project.
- Use GitHub to collaborate and code-review teammates' implementation.

At the end of semester, we'll have a competition to evaluate the performance of each team's compiler.

## 2. Schedule

| **Date** | **Details** |
|---|---|
| 28 Apr | Introduce your team in class |
| 1 May, ~11:59 pm | Submit two documents:<br>Requirement and specification<br>Planning |
| 2 ~ 15 May | Sprint 1<br>8 days: development<br>6 days: code review+revision, write progress report |
| 16 ~ 29 May | Sprint 2 |
| 30 May ~ 12 June | Sprint 3 |
| 13 ~ 15 June | Wrap up |
| 16 June | Competition |

## (1) Introduce your team in the class

Each team will have 3~5 minutes to introduce themselves in the class (Wed, 5:00 pm).

- Introduce your team & teammates
- Explain what you want to learn during the team project about software engineering
- The talk is lightweight; your talk does not need to contain technical things. You may use short presentation slides at your will.

## (2) Submit two documents before sprint 1 starts

You must submit two documents (A. requirement and specification, B. planning) before the first sprint. You should submit the document to TAs via e-mail (swpp@sf.snu.ac.kr) titled "[SWPP] Team N documents". Each document should not be more than 7 pages. You can use Korean for these documents.

A. Requirement and specification
    - Briefly explain optimizations that your team would like to implement during the 3 sprints.
    - Describe the optimizations that you are going to implement in sprint 1. For each optimization, describe the following information.
        i. Description
        ii. Algorithm in a high-level pseudo-code
        iii. At least 3 IR programs and the optimized target programs through the suggested optimization

B. Planning
    - Describe your plan for the project.
    - It must include each member's plan for 3 sprints.
    - It is allowed for one person to implement multiple optimizations in a single sprint.
    - One optimization can be implemented with your teammates.
    - One optimization can be implemented through multiple sprints.

**Adding Existing Optimization.**

LLVM has many optimizations, and simply calling these functions from your project can sometimes bring large performance improvements.

You may use existing optimizations, but in a restricted way.

For each sprint, at most *one* person in a team can add optimizations that already exist in LLVM. The person shouldn't have worked on adding existing optimization during all previous sprints.

**(3) Start sprint 1/2/3**

Each sprint consists of

- **Development phase** (8 days, Sunday - next Sunday 11:59 pm)
    - Students implement the planned features
    - Send pull requests to the main repository of the team
    - Assign reviewers.
- **Code-review & document phase** (6 days, Monday - Saturday 11:59 pm)
    - Assigned reviewers review the pull requests
    - Reviewee updates the pull request according to the comments.
    - If the update is complete, the pull request is merged into the main repository
    - The team writes a progress report of the sprint and requirements and specifications for the next sprint. They are sent to TA via e-mail.

You may use Korean in each process.

**Pull Requests.**

- A pull request has an implementation of a single feature.
- The title of a pull request should start with [Sprint N].
- It should contain unit tests to check the added functionality.
- Its line diff should be around 200 lines except comments, spaces, tests, and non-C++ code files such as .gitignore or Makefile.
    - **Once per semester**, each student is allowed to write a longer pull request (about 300 LOC).
- It should satisfy the good pull request conditions that are described in Apr. 15th's practice session
- Each pull request should be merged by 'squash and merge'.

**The Policy for Writing Pull Requests & Reviews.**

- One student can make at most 2 pull requests per sprint.
- For each pull request, 2 or more reviewers should be assigned.
- If necessary, for each iteration a team can write one pull request that does ***non-functional changes*** such as directory structure changes / source file splitting / etc.
    - Please add "[Sprint N, NFC]" in the beginning of the title.
    - This pull request needs to be reviewed but not as much as functional ones.
    - This pull request is exempt from the 2 pull request restriction.
    - The pull request can be merged during the development phase.

**Main Repository.**

- A main repository should contain a master branch.
- After each commit, the project should work correctly. TA will check

```
./configure.sh <LLVM DIR>
make -j<# cores>
make test
```

  from the root directory of the branch.

**Using GitHub Issue.**

- Please use GitHub issue to show that you are actively communicating with people.
- If there was an offline discussion and it wasn't written as comments at Pull Request, please record it at GitHub Issue.

## (4) Write & submit documents at the end of each sprint

At the end of each sprint, each team should submit a progress report and the updated requirements/specifications for the next sprint.

A. Progress report
   - Describe each member's progress at the end of the sprint.
   - If you did not finish what you have planned, explain why.
   - Include results of all the existing tests as well as your own tests to show the effectiveness of your optimizations.
   - If there is an update in planning, please submit the updated part as well.

B. Requirement and specification
   - Describe optimizations that you are going to implement in the next sprint. For each optimization, description / pseudo-code / IR programs should be included.
   - Please submit it before the next sprint.

You should submit the document to TAs via e-mail (swpp@sf.snu.ac.kr) titled "[SWPP] Team N documents". Each document should not be more than 5 pages. You can use Korean.

| | 1 May<br>11:59 pm | 15 May<br>11:59 pm | 29 May<br>11:59 pm | 12 June<br>11:59 pm |
|---|---|---|---|---|
| Requirement and specification | O | O | O | O |
| Planning | O | | | |
| Progress report | | O | O | O |

**(5) Wrap-up & Competition**

After 3 sprints, you will be given a few days to wrap up your implementation.
In this period, you may commit codes all you want (no code review, no line diff constraint, …)

On the last day, we'll run competition, estimating the correctness & efficiency of the compiler.