Appendix: Common Questions

Here are some common questions that we collected on Part 1, especially for installing environment and compiling `gem5` :

Pull image issue

If you failed to pull the Docker image with `docker pull` , it is probably because the Docker Hub is blocked currently. Try to use VPN or download the image from our PKU Disk.

Warning about `pre-commit`

If you run `scons` to compile and are warned with `Cannot find 'pre-commit'` (messages like the following), you can ignore the message. This is only required for contributing to `gem5` and will not prevent the compilation.

```
You're missing the pre-commit/commit-msg hooks. These hook help to ensure your
code follows gem5's style rules on git commit and your commit messages follow
our commit message requirements. This script will now install these hooks in
your .git/hooks/ directory.
Press enter to continue, or ctrl-c to abort:
 Cannot find 'pre-commit'. Please ensure all Python requirements are
installed. This can be done via 'pip install -r requirements.txt'.
It is strongly recommended you install the pre-commit hooks before working with
gem5. Do you want to continue compilation (y/n)?
```

Format issue (`\r`) with `git` and Windows

If you use Docker Desktop on a Windows system, you may meet a file format issue during compilation, especially when you clone the codes under Windows using `git` . `git` will

automatically replace `\n` (LF) with `\r\n` (CRLF) for you. SCons may report the occurance of `\r` (or `^M`) and throw an error. Here's three possible solutions:

- Disable the auto replacement by `git config --global core.autocrlf false`, and clone (re-clone) the codes.
- Use `dos2unix` under your container (you need to firstly install it using `apt`) to transform `\r\n` to `\n`. The command referring to this blog can be

```
find . -type f -print0 | xargs -0 dos2unix
```

- Clone the codes inside container. Be careful, **you will lose your codes when you remove the container** if you choose this solution.

Run out of memory (SCons killed) issue

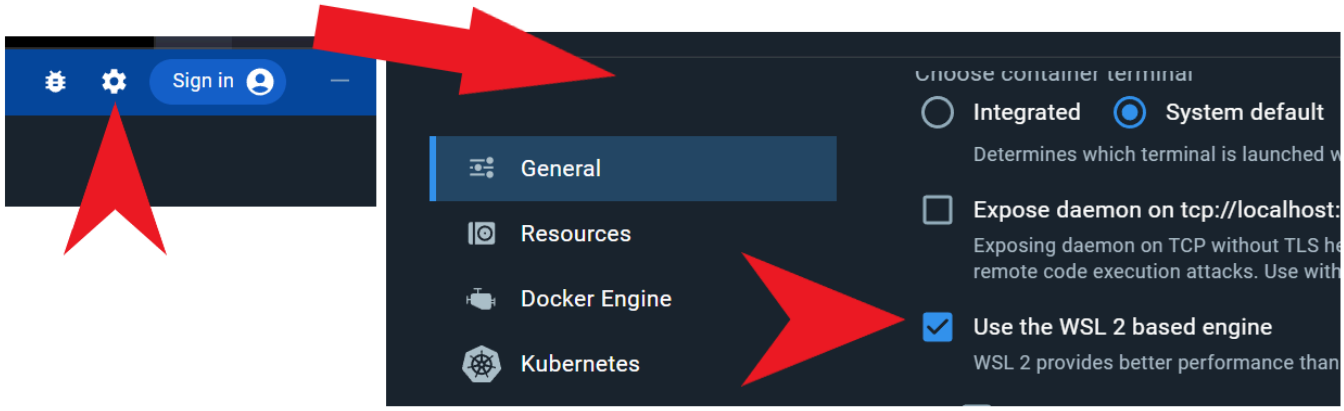
If you fail to compile and SCons reports message like the following, you are probably running out of memory:

```
collect2: fatal error: ld terminated with signal 9 [Killed]
# ... or
g++: fatal error: Killed signal terminated program cc1plus
```

As reference values, with 8 compile jobs (`-j 8`), the compilation and the link process (`[LINK] -> build/ARM/gem5.opt`) may both take up to ***12GB*** memory. If your PC's physical memory is equal to or less than 16GB, you may probably meet this issue.

If you face this trouble, you can manually enlarge the swap space used by Docker container. On Windows, you need to firstly change the backend to WSL2 (installation method can be

found in official doc). The setting process is like the following:



Next, stop Docker Desktop, and stop WSL by `wsl --shutdown`. Then, create a file named `.wslconfig` under `C:\Users\Yourname`, and write the following configs:

```
[wsl2]
memory=4GB
swap=32GB
autoMemoryReclaim=gradual
```

You can adjust the `memory` and `swap` values based on your PC's physical memory and disk space limits. WSL2 uses up to 50% of your physical memory by default. According to the memory usage values described above, the `swap` is recommended to be larger than ***20GB***. Set `autoMemoryReclaim=gradual` to save your physical memory when you are not running WSL or containers.

Lastly, start WSL, Docker Desktop and your container again. You can also start a new container by restricting its memory and swap limits:

```
docker run -it --memory 4G --memory-swap 20G -v ...
```

Remember to use the same `<container path>` in `-v`, otherwise compilation may fail. Or you can delete `build/` and re-compile completely.

Use docker on server

If you still cannot compile or cannot setup WSL and Docker Desktop, feel free to contact TA for a server account. However, please note:

- You still need to **use docker image**, since the environment on the server is not checked.
- You **DO NOT need to pull** the image again. Just clone the codes and use `docker run`.
- Because you can see all existing containers on the server, please assign an ***UNIQUE NAME*** to your new container with `--name` in `docker run`. If you use a name same with an existing one, you will not be able to start the new container.
- **DO NOT MALICIOUSLY ENTER OTHER STUDENTS' CONTAINERS OR MODIFY THEIR CODES!**