

WACC compiler project

Group WACC19

Ayman Moussa (am5514)
Henryk Hadass (hh2214)
Nana Asiedu-Ampem (na1814)
Oliver Wheeler (ow14)

December 15, 2015

The Product

Over the past month we have been stably developing our compiler from the bottom up. Although we faced several hindrances and obstacles along the way, we were able to achieve a fully functional working compiler that produces ARM assembly as expected. As it stands now, it is possible for our compiler to be further developed into a more powerful complex compiler that supports a wide range of features that is currently unsupported such as optimisation, dynamic variables etc.

Add more more stuff here

The Product Management

Initially the team was split in the sense that each individual was assigned to a task to complete in order to maximise the throughput of the task:

Frontend Oliver created the Lexer, Henryk created the Grammar, Nana created the parser with Ayman helping him doing so...

Backend Oliver restructured the abstract syntax tree with fixes. Henryk, Nana and Ayman were mainly responsible for producing assembly output by walking the abstract syntax tree.

Throughout the whole project, we utilised the version control system Git in order to manage our project. This proved extremely useful as there were times when we made changes to code which caused tests to fail but by using Git we were able to see which changes were made on which file and by who. Changes were then reverted to working code and the solution to the problem was proposed and pushed to Git.

We faced several hindrances near the end of the project:

- The reference compiler everyone was using went offline for almost 10 hours which left us unable to proceed as we depended heavily on the reference compiler for a model output of assembly for our test cases.
- Our code was apparently producing a non UTF-8 character which caused the auto-tester to crash and produce a two-page report and not display any of the other errors in our test cases. This impacted on our development as we needed to see exactly which test cases were failing and why but were unable to see them because of this problem. Making changes to code that fixed one problem but caused this problem had to be reverted several times to old broken code just to see our test case errors. It was not only our group who was faced with the problem but we believe we could have achieved more if the original problem was fixed in the first place.
- "Supposedly fixed" broken code that was the source of problems for other team members as they were wondering why their test cases were failing...

The Project Management: An analysis of the organisation of your group and your use of project management tools (such as Git). You should describe how your group was structured, how you coordinated your work and detail any tools that helped/hindered your progress. You should also discuss what went well and what you would do differently if you were to do the lab again.

The Design Choices The Design Choices: An analysis of the design choices that you made during the WACC lab. You should discuss the design patterns you used when designing your code and why you chose to use them.

Beyond the Specification

Our WACC compiler (formally WACC++) now supports a range of features which extend the original WACC specification thus allowing us to create more powerful programs at our disposal:

WACC++ features A

- **Classes:**
- **For loops:**
- **Side Expressions:**

Optimisations Optimisations made during the WACC++ compiler development were ...

Future extensions Preferably we would have liked to implemented features such as ... but given the time constraints we would not have wanted to rush our compiler development.

Beyond the Specification: An evaluation of your extensions to your WACC compiler. You should describe all of the language extensions, optimisations or other aspects that you have added to your compiler, including how these features can be accessed or viewed. You should also brievely discuss what future extensions you would like to add to your WACC compiler if you had more time.

Spare Maybe let's write what we learnt and gained from this group project.