Requirements document software.

Team “Gold Compilers”

Compiler Foxy.

Contents.

Project Information:

Company or Organization:	Gold Compilers
Project:	Compiler
Project star date:	10-July-2021
Project end date:	13-August-2021
Client:	Ing. Norberto Jesús Ortigoza Márquez
Project sponsor:	Ing. Norberto Jesús Ortigoza Márquez
Project manager:	García Felipe Miguel

Approvals:

Name:	Date:	Job:	Observations:	Sign:
Felix Flores Paul Jaime	13-August-2021	Tester	The compiler work's correctly	
San Juan Aldape Diana Paola	13-August-2021	The System Integrator	The compiler work's correctly	
García Felipe Miguel	13-August-2021	Project Manager	The compiler work's correctly	
Norberto Jesús Ortigoza Márquez	13-August-2021	Client		

Purpose & scope:

Compiler for c According to what was discussed with the client, a compiler for C, which in this first version can compile a file that returns a single “integer” number.

- Version 1:
  - Our compiler must can return an integer, compiling simple fragment code.
- Version 2:
  - Our compiler must can return a result operated with unary operations like negate result, positive, bitwise, and logical negation.
- Version 3:
  - Our compiler must can operate a binary operation like sum, subtraction, multiplication and division.

According to what the client requires, we will make a compiler, to achieve the scope of the project we will create a compiler which is capable of compiling elements "Integer", unary operators, as well as operations between them.

To reach this proposed scope, we will base ourselves on the tutorial by “Nora Sandler”, until its fourth installment, which the client has provided us with.

We will also rely on several courses on how to use git, elixir programming and our knowledge in assembler.

Features.

Our product requires users with experience in at least the language of c programming and use of the UNIX operating system

- You will be able to use the compiler foxy in a basic way with its minimum functions, for this you will need experience in basic use of both cases UNIX and C programming language.

- You will be able to make structural changes to improve the compiler code and use it to its fullest capacity, also requires basic knowledge of how compilers work, as well as the parts that compose it.

Commands that we use in the project:

git init	Git init is one way to start a new project with Git. To start a repository, use either git init or git clone - not both. To initialize a repository, Git creates a hidden directory called. git. That directory stores all of the objects and refs that Git uses and creates as a part of your project's history. .
git clone	Git clone is used to copy an existing Git repository into a new local directory. The Git clone action will create a new local directory for the repository, copy all the contents of the specified repository, create the remote tracked branches, and checkout an initial branch locally.
git add -A	stages all files, including new, modified, and deleted files, including files in the current directory *and* in higher directories that still belong to the same git repository
git commit -m "Descriptive text"	creates a commit, which is like a snapshot of your repository. These commits are snapshots of your entire repository at specific times. You should make new

	commits often, based around logical units of change.
git push	Git push updates the remote branch with local commits. It is one of the four commands in Git that prompts interaction with the remote repository. You can also think of git push as *update* or *publish*.
git pull	git pull updates your current local working branch, and all of the remote tracking branches. It's a good idea to run git pull regularly on the branches you are working on locally. Without git pull, (or the effect of it,) your local branch wouldn't have any of the updates that are present on the remote.
git branch <branchName>	List all of the branches in your repo. Add a <branch> argument to create a new branch with the name <branch>.
git status	When in doubt, run git status. This is *always* a good idea. The git status command only outputs information, it won't modify commits or changes in your local repository.

How we work?

The compiler will be developed with 64-bit architecture our main operating environment and where the compiler works correctly will be UNIX (Ubuntu).
For the development of the compiler, we will use the github.com platform to be able to control the versions of the software to be developed.

As the main means of communication, we will use the application “WhatsApp”, Facebook and GitHub, the physical meetings will be canceled because we cannot meet in person.

Knowledge learned:

Members:	Conclusions:
García Felipe Miguel 	The project was concluded satisfactorily, in the process of realization we had a large part of complications, but with the help of the teacher and researching in books we were able to solve each one of them and be able to continue developing our code. To understand the code that the teacher gave us, we had to spend a lot of time, since it is very complex and understanding line by line is usually a bit difficult. To design the different tests that our compiler must pass, my team and I first set ourselves the task of understanding how the compiler works from the general to the more specific and thus knowing which tests it had to pass and which ones did not. For the use of the git platform at first it took me a bit to get familiar since I had not used it much before, but as the project progressed and commits.
Felix Flores Paul Jaime 	In this project I have understood how a compiler works, it is very interesting to me how it is formed from the beginning, at the beginning it was difficult to understand it because it is a very abstract topic and not so easy to digest. Thanks to Nora's document I have done a lot easier as a compiler really works. The compiler is the cornerstone when you want to create any operating system. The teamwork was quite good, if anyone had doubts with any part of the code among all the members, we helped each other so that no one had doubts and thus have a better performance. Finally, the delivery of our work was concluded in a timely manner.

San Juan Aldape Diana Paola 	When carrying out this project I was able to verify that many concepts are required to understand the operation of compilers and with this their realization, mainly concepts of formal languages and automata, since lexical and syntactic analyzers are a fundamental part of compilers. As well as knowledge of the structure of computers since it was necessary to have cognition of how the language of the machine works. Finally, I consider that the parser is the most important part in the compilation process, which is reflected when implementing the code.

We hope that this compiler will be easier and more understandable for you, so in the same way, we hope that the user manual is as understandable as possible and that you will be happy with our work.

Thanks' for using the Foxy compiler.

References.

Type:	References:
Website	Norasandler.com. *Writing a C Compiler, Part 1*. Website: https://norasandler.com/2017/11/29/Write-a-Compiler.html [Accessed 27 July 2021].
Website	Norasandler.com.*Writing a C Compiler, Part 2*. Website: https://norasandler.com/2017/12/05/Write-a-Compiler-2.html[Accessed 27 July 2021].
Website	Norasandler.com.*Writing a C Compiler, Part 3*. Website: https://norasandler.com/2017/12/15/Write-a-Compiler-3.html [Accessed 27 July 2021].
Website	Course elixir: https://www.youtube.com/c/makigas/videos
Website	Course github: https://youtu.be/FZRLRVxlxUw [Accessed 27 July 2021].
Website	Test for c: https://github.com/nlsandler/write_a_c_compiler/tree/master/stage_1 [Accessed 27 July 2021].
Website	Test for c: https://github.com/nlsandler/write_a_c_compiler/tree/master/stage_2 [Accessed 27 July 2021].
Website	Test for c: https://github.com/nlsandler/write_a_c_compiler/tree/master/stage_3 [Accessed 27 July 2021].

Historic:

Author	Description	Version	Date
San Juan Aldape Diana Paola	First Version	1.0	26-07-21
San Juan Aldape Diana Paola	Second Version	2.0	13-08-21