BRIGADA DE LOS AZTECAS.

COMPILER.

Who are we?

Brigada de los aztecas is a group of systems engineering students of the Engineering Faculty of the Universidad Nacional Autónoma de México prepared to develop efficient and quality software.

Mission.

Build technological solutions to the problems that the society pass through.

Vision.

Our team seeks to position as a solid and productive company with a great importance in the technology area, through evolution and innovation.

Table of contents.

I. Objective. 4

II. Introduction. 4

III. Project requirements. 5

IV. Work plan. 7

V. Architecture. 12

VI. Test 14

VII. Conclusions. 15

VIII. Appendix. 16

I. Objective.

The students should be able to plan, design and implement a project where it will develop a compiler for the C language, using the concepts seen in the Compilers class, imparted by the Engineer Norberto Jesús Ortigoza Márquez.

II. Introduction.

In the present project we will apply through a dynamic and functional language known as Elixir, the construction of a compiler for the C language, likewise we will lean on the notes and suggestions of Nora Sandler´s investigation.

The work is divided in 4 deliverables, in one hand, it starts setting the scope of the compiler, as well as the requirements that we must done.

During the first delivery we will find the elaboration of lexer which is also known as scanner, the parsing, the construction of the automated testing and the identification of the number of lines so its easier to find errors of any type in the code.

III. Project requirements.

The project to develop is a compiler of the C programming language, for an architecture of 64 bits, developed in the Elixir programming language.

The project will be divided in 4 deliveries during the semester 2021-1 where we will have to add some functions based on the next points.

- On the first delivery the compiler should be able to recognize characters used in the C language, so that must be able of read the next function.

  int main ( ) {

  return 2;

  }

- To the second delivery the compiler shall support the unary operators, which are:

- Negation (-).
- Bitwise complement (~).
- Logical negation (!).

- On the third delivery our compiler shall support the next operations.

- Sum (+).
- Subtraction (-).
- Multiplication (*).
- Division (/).

- On the fourth and last delivery our compiler shall support the next binary operators:
- Logical AND (&&)
- Logical OR (||)
- Equal to (==)
- Not equal to (!=)
- Less than (<)
- Less than or equal to (<=)
- Greater than (>)
- Greater than or equal to (>=)

At the end of the last delivery the compiler must be structured by the following modules.

- Scanner.
- Analyzer.
- Code generator.
- Optimizer (optional).
- Besides that, it must support the next function.

```
int main() {
    return (3+4 <= 4 || 1&&2 != 3 > -6);
}
```

For the presentations every single member of the team must expose, and this presentation will last 20 minutes.

IV. Work plan.

Stakeholders list.

- Project Manager.
  Cruz Ramírez Cesar Alejandro

- System integrator.
  Higareda López Carlos Alberto

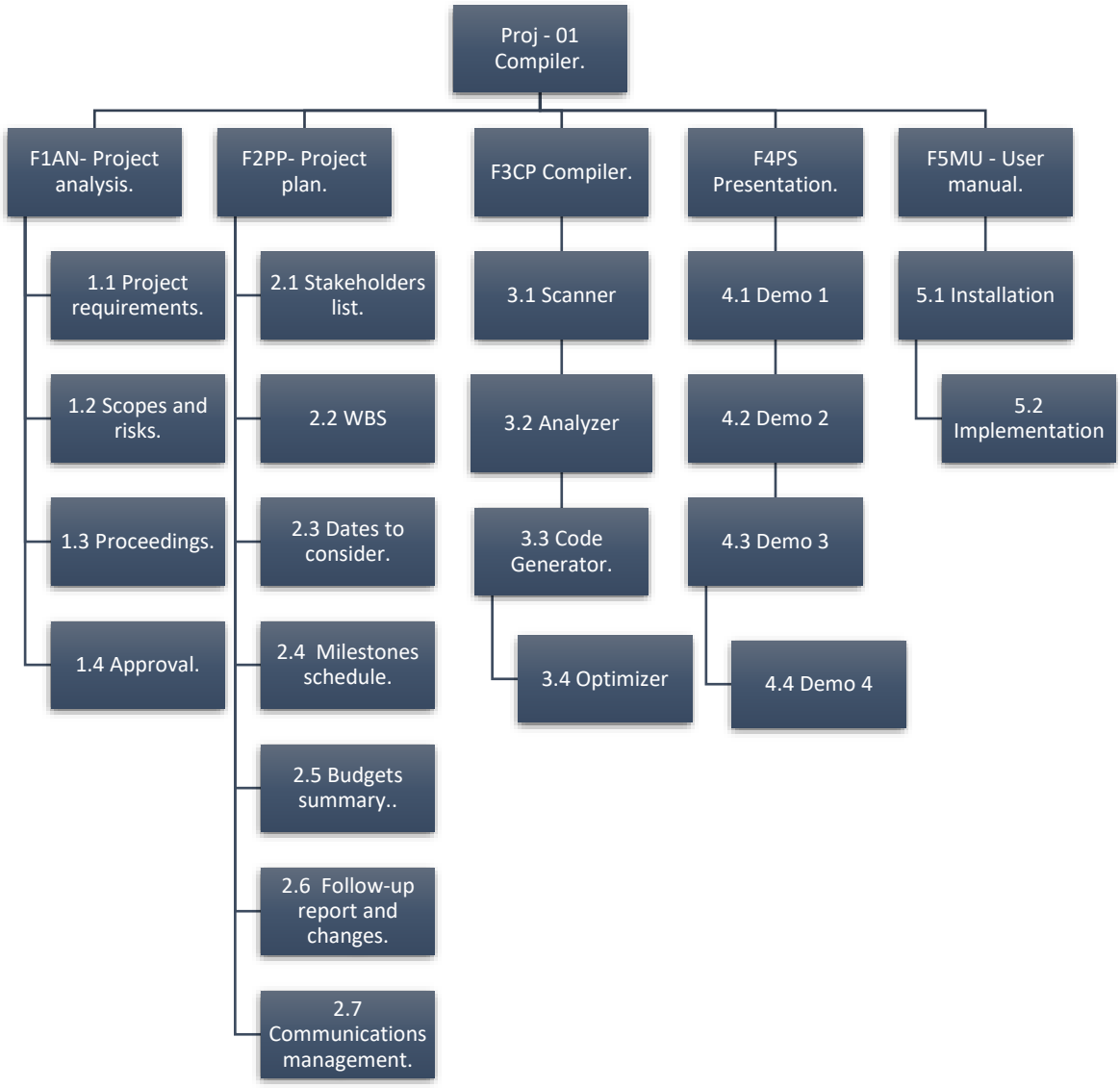
- System architect.
  Peralta Correa José Roberto

- Tester.
  González Ochoa José Antonio

- Developer.
  Leyva Pérez José Luis

- Sponsor.
  Ing. Ortigoza Márquez Norberto Jesús.

WBS.


Proj - 01 Compiler.
F1AN- Project analysis.
F2PP- Project plan.
F3CP Compiler.
F4PS Presentation.
F5MU - User manual.
1.1 Project requirements.
1.2 Scopes and risks.
1.3 Proceedings.
1.4 Approval.
2.1 Stakeholders list.
2.2 WBS
2.3 Dates to consider.
2.4 Milestones schedule.
2.5 Budgets summary..
2.6 Follow-up report and changes.
2.7 Communications management.
3.1 Scanner
3.2 Analyzer
3.3 Code Generator.
3.4 Optimizer
4.1 Demo 1
4.2 Demo 2
4.3 Demo 3
4.4 Demo 4
5.1 Installation
5.2 Implementation

Dates to consider.

- September 22, 2020 – Start of classes.
- October 29, 2020 – First delivery.
- November 26, 2020 – Second delivery.
- January 24, 2021 – Third delivery.
- January 28, 2021 – End of classes.
- February 11, 2021 – Deadline for the fourth delivery.

Milestones schedule.

Activity	Duration	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16
Negotiation	12 horas.																
Requirements	2 horas.																
Scopes and risks.	4 horas.																
Roles appointment.	2 horas.																
Proceedings lift.	2 horas.																
Approval.	2 horas.																
Project plan.	112 horas,																
Development of the first delivery.	16 horas.																
Development of the second delivery.	16 horas.																
Development of the third delivery.	16 horas.																
Development of the fourth delivery.	16 horas.																
Delivery documentation.	32 horas.																
Deliveries presentation.	16 horas.																

Budgets summary.

Category.	S1-S4	S5-S8	S9-S12	S13-S14	
Salaries.	$50 000	$50 000	$50 000	$50 000	
Tools.	$50 000	-	-	-	Total.
Services.	-	$500	-	$500	$251 000

Communication management.

Stakeholder.	Communication.	Canal.	Frequency.	Motive.
Sponsor.	Formal meeting.	Zoom.	Once per delivery.	Project tracking.
PM.	Formal meeting.	Discord.	Once per week.	Project tracking.
System integrator.	Formal meeting.	Discord.	Once per week.	Project tracking.
System architect.	Formal meeting.	Discord.	Once per week.	Project tracking.
Tester.	Formal meeting.	Discord.	Once per week.	Project tracking.
Developer.	Formal meeting.	Discord.	Once per week.	Project tracking.
Work team.	Informal messaging.	Discord.	Indefinite continuous.	Project development.
Work team.	Formal collaboration.	GitHub.	Indefinite continuous.	Project development.
Sponsor.	Informal messaging.	Discord.	Indefinite continuous.	Project tracking.

Change request.

Change request.

Brigada de los aztecas compiler.

Change request number.	1.
Change applicant.	Cruz Ramírez Cesar Alejandro.
Applicant area.	Project management.
Place.	Engineering faculty.
Project sponsor.	Eng. Ortigoza Márquez Norberto Jesús.
Project manager of the project.	Cruz Ramírez Cesar Alejandro.

Change category.

- Schedule.

Cause and origin of the change.

- Work team request.

Description of the proposal change.

- It is proposed a postponement of the first delivery of the project due to technical difficulties that the work team has, the proposal for the first delivery of the project is November 27, 2020.

Project Sponsor.

Project Manager.

V. Architecture.

The compiler is conformed by four modules, as suggested by Nora Sandler, these are Lexer, Parser, Code Generator and Linker.

Sanitizer

Cleans the code received before starting in the compilation process, that means that this remove the spaces and the line breaks, for a better management in later stages and avoid possible errors.

Lexer

Lexer will oversee the generation of the tokens from the chain provided by the sanitizer. These tokens will be obtained using a grammar.

For the first phase of the compiler the tokens that will be supported are:

- open_brace "{"
- close_brace "} "
- open_paren"("
- close_paren ")"
- Semicolon ";"
- return_keyword "return"
- int_keyword "int"
- main_keyword ‘'main’'

Parser

Generates an AST tree from the tokens obtained by the lexer. This tree represents its grammar structure. Because this is done from rules, we do an error analysis in the code while this generates the AST tree.

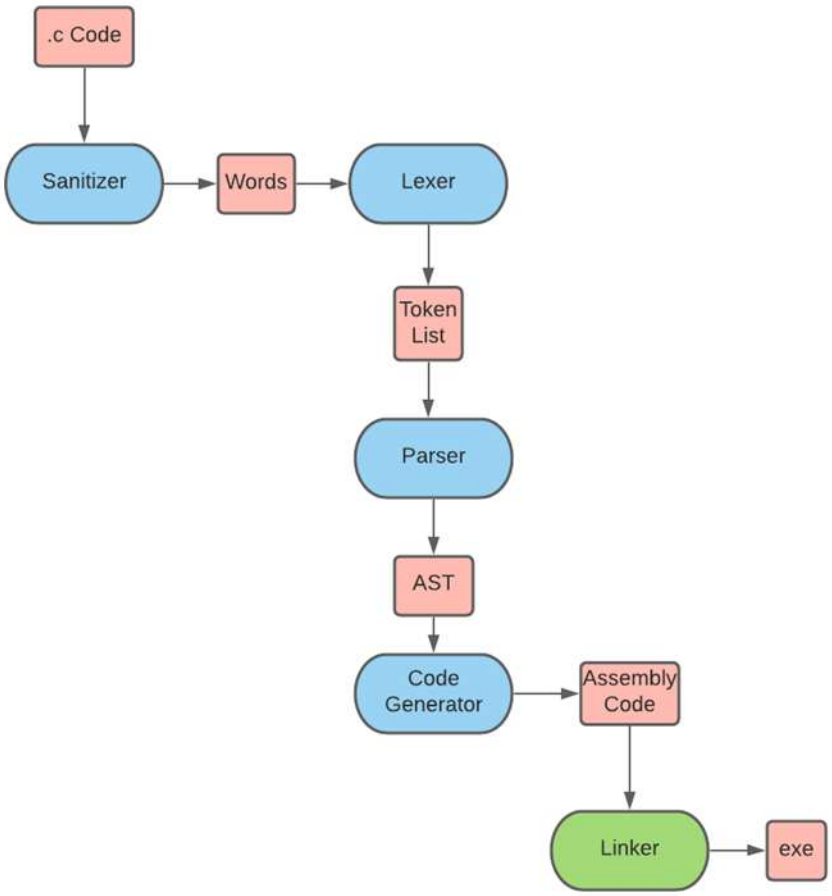
Code generator

This generates the instructions in the assembler from the AST tree previously obtained.

Linker

This takes the assembler code and with gcc help generates the executable.

It can be seen the architecture of the project in the following diagram, showing the connections in the modules.



VI. Test.

For the first delivery the tests provided by Nora Sandler were done, where six of them were valid and the other six were invalid. We do them manually and automatically. The compiler passed the test phase because we obtained the expected results.

The tests of the first delivery were the following.

Valid tests	Invalid tests
multi_digit.c	missing_paren.c
newlines.c	missing_retval.c
no_newlines.c	no_brace.c
return_0.c	no_semicolon.c
return_2.c	no_space.c
spaces.c	wrong_case.c

VII. Conclusions.

VIII. Appendix.

The compiler of the Brigada de los Aztecas is found in a GitHub´s repository, so that we need to consult the following link, to download the .zip folder or clone the repository in our computer through Visual Studio, terminal or the IDE of your preference.
https://github.com/hiphoox/c211-aztecas.git