Fakultät Informatik

Development Report about Ion shell for Redox Os

It Project Report im Studiengang Informatik

vorgelegt von

Florian Naumann

Matrikelnummer 3528558

Betreuer: Prof. Dr. Christian Schiedermeier

Contents

0.1 Abstract

This report describes the development contribution to Ion Shell for the It project "redox" between the winter term 2022 and summer term 2023. First an introduction about Ion Shell is provided. In the next chapter the typical development process which I applied, is shown via examples of bug fixes and feature implementations for Ion Shell. In additions that section indents to showcase how documentation, testing and coding can be performed for Ion Shell. After that chapter a reflection outlines the current status of Ion Shell. This reflection also involves achievements of my development work and left tasks/goals I could not complete. In the last chapter a listing of all my created issues and conducted pull requests are provided to quantify the development work.

0.2 Abstract in German

Dieser Bericht beschreibt die Entwicklung als Beitrag zu Ion Shell für das It Projekt "redox" während dem Wintersemester 2022 und Sommersemester 2023. Zuerst kommt eine Einleitung zu Ion Shell. Im nächsten Kapitel wird der typische Entwicklungsprozess, den Ich angewendet habe, durch Beispiel in Form von Fehlerbehebung und Implementierung von Features aufgezeigt. Zusätzlich soll dieses Kapitel demonstrieren wie Dokumentation, das Testen und die Programmierung für Ion Shell durchgeführt werden kann. Nach diesem Kapitel soll der aktuelle Status von Ion Shell verdeutlicht werden. Dieser Teil soll auch die Errungenschaften meiner Entwicklungsarbeit und die übrigen Aufgaben/Ziele welche Ich nicht vollendend konnte, aufzeigen. Im letzten Kapitel wird eine Auflistung von all meinen erstellten Issues und durchgeführten Pull Requests dargestellt um meine Entwicklungsarbeit zu quantifizieren.

0.3 Sources of code snippets

In several places, code snippets are used to illustrate certain points. Code snippets are taken from the Ion Shell repository on Gitlab [1]

based on the commit id 60bfb73351f0412c95b8ba2afe75e988514470a6. Unfortunately I could not generate an online link to a file based on a certain commit. For this reason I include a local copy of the repository beside this report. The name of this snapshot is "ion.zip". This local copy is a snapshot of the repository based on the commit 60bfb73351f0412c95b8ba2afe75e988514470a6. A source item in the bibliography of a code snippet is therefore a local file path within this local snapshot.

The Notation for file paths is as follows:

“‘ <folder> -> <folder> -> .. -> <file> “‘

Therefore the following Example:

“‘ src -> shell -> flow.rs “‘

Equals the flow.rs file inside the folder shell which is inside the folder src at the root project.

Chapter 1

Introtuction

1.1 What is Ion Shell

Ion Shell is a program written in rust. It is maintained on a git repository on the gitlab server for redox os ecosystem [2]. It is a shell which executes commands within a terminal emulator via read evaluation loop. Like other shells, Ion Shell also allows the execution of scripts in its own language. It serves as the default shell on redox os. As moment of writing ion shell can also be used on Linux. This shell is interpreted statement by statement. There is an online manual for the usage of ion shell available [3].

The work which was performed during the it project resolved mainly around the scripting aspect of Ion Shell. Because of this the scripting language of Ion Shell is introduced in more details as a preamble. To better illustrate certain aspect of scripting with this shell, certain differences between It and bash are discussed.

1.1.1 Typing in Ion Shell

As an example Bash only operates on text and does have not have concept of types. Ion Shell on the other hand works in structural typing. A certain section of the online manual of Ion Shell lists all possible types [4]. Form this reference we conclude that Ion shell has following primitive types:

- str - bool - int - float

And also provides these structural typing via:

- Array: - Hashmap: - Btreemap:

Working with these types instead of just text provides a better detection of programming errors. Since Ion Shell is interpreted statement by statement, these errors are caught at runtime though.

It is important to note there is no nominal typing in Ion Shell. Formal Declaration of objects or structs with named fields is not possible like in other programming languages like Rust, Java, C# etc ..

To better illustrate how scripting in Ion Shell looks like a few core features are explained with code snippet as examples.

1.1.2 Expansion

Ion Shell also supports expansion functionallity.

1.1.3 String and array methods

There are builtin utility functions avaible for scripting called methods. There are 2 kinds of methods. The kind of method is determined by the type of the return value. Every kind of method is prepended with its own sigil

- String methods. Prepended with a "$" sigil. Returns a string. See 1.1 as an example.
- Array methods. Prepended with a "@" sigil. Returns an array. See 1.2 as an example.

```
# Lines splits a string into an array of elements.
# These elements are seperated by "\n" as new line.
echo @lines($unescape("firstline\nsecondline"))
for line in @lines($unescape("third\nfourth\nfifth"))
    echo $line
end
# Output
# firstline secondline
# third
# fourth
# fifth
```

Listing 1.1: Example of a string method. Source: [5].

```
# Returns a string which the repetition of the 1. argument
# times 2. argument as a number.
echo $repeat("abc, " 3)
# Output:
# abc, abc, abc,
```

Listing 1.2: Example of a array method. Source: [6].

1.1.4 Control Flow

Chapter 2

Development in Ion Shell

2.1 Finding areas for contribution

With the goal to contribute to Ion shell during the period of the it project, the possible bug fixes/features needed to be determined first. The following activities were undertaken to find out concrete tasks to accomplish:

- Going through the online manual of Ion Shell. - Reading posted issues on the GitLab Repository of Ion shell.

The online manual of the shell is meant for the users of the application. It focuses on the following points:

- What and how something can be done in an interactive session. - Features of the scripting language. - Philosophy the application. Why does it exits and what it is supposed to be.

Investigating the online manual of Ion shell provided the overview of already implemented features and how the shell was meant to be used. While reading the manual, it became clear that the documentation lacked explanations in certain areas. Later on additions to the documentation were also required for new implemented features in Ion Shell. Because of this the manual presents one of the fields in Ion shell for contribution.

Another important source for the deduction of tasks is the collection of issues and pending pull requests from a repository. This investigation yielded a number of goals to undertake. However it also pointed out that the development of this project stagnated for a while. Over the course of the it project I would be the only contributor, most of time.

2.2 Automated testing of features

Unit tests and integration tests are the 2 forms of tests used to verify the functionality of the shell. The observation of the code base suggests that the first form as unit tests are mainly used to inspect the correctness of tokenisation, parsing and evaluation of the builtin functions of the shell in isolation. It is important to note that these unit tests execute and assert inner/private functions too in contrast to other programming languages like CSharp where unit test should not access private functionality easily. This is typical in the Rust language because unit tests are often in the children module in respect of the code to be tested. Visibility in Rust enables code in a children module to have full access to all functionality in a parent module. This allows to test private functions in Rust easily [7].

Integration tests vary depending on the kind of application. Ion shell's approach to integration test is the execution of script files and comparing the output to text files which contain the expected output for the test. These testing does therefore focus on the scripting aspect of the shell. Especially this kind of testing is valued highly according to the Contributing Guidelines of this porject [8]. Since rust does not provide a builtin way to perform these kinds of tests, a rather complex bash script was written to orchestrate this kind of integration tests. Before this it project there was no explanation of how these new integration tests are executed and how they should be written. It was however paramount to add new integration tests to verify bug fixes and implemented features. A documentation section in the contributing guideline was crafted during the process of learning the working of this integration tests [8]. This will hopefully reduce the learning effort for future contributors in the area of integration tests.

2.3 Example of development

2.3.1 Implementing an array method in Ion Shell

The feature Parameter Substitution via the array method "subst" ought to represent in the development work in Ion Shell. The following tasks were performed to archive the implementation:

- Determine the desired behaviour of the array method. - Create a test to verify the functionality of the implementation. - Implement the method in Ion shell. - Document this feature in the online manual of Ion shell.

The implementation was proposed and merged under the Pull Request [9].

2.3.1.1 Finding out the desired behaviour of "subst"

This feature was requested in an issue a certain while ago [10]. In this issue the specifics of behaviour of this array method have already been discussed too. The discussion concluded the following signature of "subst".

“‘text subst(input: T, default: T) -> T “‘ source: [11]

Where the invariant says that T must be of type array. This array method takes two arguments as arrays and returns either the first or second argument. The first one is returned if it is not an empty array. Otherwise the second argument is returned. The accomplished implementation follows the signature and invariant.

2.3.2 Creating integration test for the implementation

To verify the functionality of the implementation, an integration test was added to test suit. An integration test consists of 2 files. One is an ion file which is executed by the shell an other is a plain text file with extension "out". This text file describes the expected output of the execution of the ion test file. For this implementation there is the file "subst.ion" and "subst.out" where "subst" is the name of the test. Both file must share the same base file name, here "subst".

The code snippet 1.1 represents the test code.

```
1
2  let array = []
3
4  # Inline array in the method
5  echo @subst(@array [foo bar])
6
7  # single value
8  echo @subst(@array [baz])
9
10 # variable expansion
11 let default = [foobar]
12 echo @subst(@array @default)
13
14 # method would not trigger
15 let array ++= faz
16 echo @subst(@array @default)
17
18 for number in  @subst([] [2 3])
```

```
19    echo $number
20  end
```

Listing 2.1: integration test for subst method. Source: [12]

2.3.3 Implementation of "subst"

2.3.4 Documentationa of "subst"

Chapter 3

Conclusions after the development

3.1 Accomplishments

Over the period of the it project, participation in a number of areas of development were performed:

- Improving Code quality for example refactoring.
- Documentation.
- Writing unit tests and integration tests.
- Bug fixes.
- Implementation of features.

It was possible to get the code quality to a point where no warnings or complication errors occurred. The all pull requests described in the section 4.1.3 contributed to this success.

Contribution in regard of documentation was another portion of development, listed in 4.1.4. This not involved the markdown files at the project root and the comments within the source code but also the part of the ion manual. This will hopefully server future contributors well.

Ion Shell as any other software contains bugs of course. Some the bugs could be fixed via code contribution on my side, listed in 4.1.2. A few features were also achieved, listed in 4.1.1.

The integration tests of Ion Shell were analyzed and comprehended. On one hand this knowledge could be leveraged to create effective tests which verify the correct behaviour of the implemented features and bug fixes. On the other hand this understanding lead to a section about how to write the integration tests in Ion Shell [13].

3.2 Status of Ion Shell

Ion Shell is not only capable of running on redox os but also quit well on a Linux distribution. This fact became more than clear by the my conducted development of Ion Shell on a Linux distribution.

The scripting API of Ion Shell allows for writing more correct automation compared to shells like bash shell. Structural typing and various higher level language features like expansion 1.1.2 methods 1.1.3 and control flow 1.1.4.

3.2.1 Problems with Ion Shell

However Ion Shell has unfortunately several troublesome areas. This section points out those aspects of Ion Shell which should be improved on.

- **Ion Shell has no regular contributors**
- **Ion Shell CI is not functioning correctly**
- **Ion Shell has no distribution for end user on Linux**
- **Ion Shell has not mainted dependencies**

Ion Shell has no regular contributors

The Gitlab repository of Ion Shell does not indicate an active development community from the beginning of the it project up to the moment of writing. The last commit before my first contribution was .. . Up to the moment of writing most commits were done by me. A few exceptions were occasional pull requests ... Throughout the it-project there was only one person who reviewed and accepted pull requests. This individual in question is , , the .. . The lack of regular contribute could lead to a complete hall of the development of Ion Shell.

Ion Shell CI is not functioning correctly

A number of dependencies used by Ion Shell are not maintained anymore ... If newly needed features or bugs arise from those libraries, then someone has to do it on their own. This puts more complexity on the new potential contributors as they might have to learn and adjust separate code bases.

Ion Shell has no distribution for end user on Linux

Ion Shell can be still considered an alpha program. During the development phase of it project I discovered pretty fast new bugs and needed features. With a larger user

space these issues could be detected early. More users could provide more feedback to the program and even attract potential contributor. Additional contributors is what this project really needs the most. Ion Shell can be complied and run on Linux. However it lacks official distribution channels for linux end user . A user indenting to use this shell, has to install the rust tool chain before and then compile the program. More Linux use rend users could be reached the program would be released on distribution channels like dnf, apt etc .. .

Ion Shell has not mainted dependencies

Not all jobs of the Ion Shell CI' do still not complete successfully. While the jobs .. now run successfully due to my contribution The other jobs .., still fail. This is problematic. While local compilation and manually run test suit can provide detection of problems, it not enforced. Every pull requests therefore require also inspection of the compilation and a working test suit for the code reviewer. This could be automated with a proper CI. Beside causing more work for the reviewer, it also can hinder the confidence of new contributors not knowing that CI can run successfully right now. I tried communicating this problem via following issues .. . In addition I talked to the community about this on the matrix server on several occasions.

3.3 What is left to do in Ion Shell

This section describes which goals for the it project could not be archived. The description also contains the reasons why a goal could be accomplished.

- Incorporate the source location, (file name, line number, column number) into the error message.
- Implement wild card expansion for various schemes like disk ..
- Make the CI work on all jobs possible.

The following bullet points were non-goals during the it-project. Nonetheless they were identified as worthwhile endeavours.

- Provide various deploy set ups to publish Ion Shell on various package managers like apt, dnf, ..
- Port Ion Shell to windows.

3.3.1 Enriching error messages

One area of contribution was focused on enriching error message in Ion Shell. Error messages in Ion Shell already provide specific information about what problem occured. However the details within certain error messages had some potential for improvement. The following merged pull requests of mine were focused on that areas:

- Out of bounds errors show the invalid index and the length of an array [14].
- Errors because of the reference of an undeclared variable now show the name of the undeclared one [15].

Another aspect of missing information for error messages was the source location where an error was encountered. The source location is the respective line and column number. The source location in an error message as a feature were noted down in an still opened issue [16]. The parser of ion shell does not keep track of line and columns numbers for statements. Because of this, one can not simply incorporate the source location into the error messages.

A workaround was attempted via counting at least the line numbers per executed statement. There is a for loop, see code snippet 3.1 at line 9, which terminates, parses and executes statements one by one. This workaround could at least provide the line number of the source location. That attempt however did not work for all scenarios. In cases of calling a function in Ion Shell wrong line numbers occur however. This oddity comes from the fact that statements within a function are executed in a batch. Let us consider the following code snippet TODO add code snippet. If an error occurs at line .. then the line number is 4 with the workaround because that is where the function was called. In addition to that the execution of builin methods do propagate their error up to the execution loop 3.1. If an error happens a message is directly printed to the console via this section TODO.

In my view the solution is to rewrite the tokenisation and parsing in a way so that a token/language items are linked to its respective column and line number. This rewrite however requires coordination with the respository owners and significant amount of work which exceeds the available time I could afford to spend for the it project.

```
1 /// Receives a command and attempts to execute the contents.
2 pub fn on_command(
3     &mut self,
4     command_to_execute: impl Iterator<Item = u8>,
5     set_cmd_duration: bool,
6 ) -> std::result::Result<(), IonError> {
7     // ... some code before
```

```
8       for stmt in command_to_execute.batching(
9       |cmd| Terminator::new(cmd).terminate()) {
10          // Go through all of the statements and build up
11          // the block stack
12          // When block is done return statement for execution.
13          for statement in StatementSplitter::new(&stmt) {
14              let statement = parse_and_validate(statement?)?;
15              if let Some(stm) = Self::insert_statement(
16                &mut self.flow_control, statement
17                )? {
18                  self.execute_statement(&stm)?;
19              }
20          }
21      }
22      // ... some code after
23      Ok(())
24  }
```

Listing 3.1: For loop which terminates,parses and interprets statements. Source: [17].

Chapter 4

List of development accomplishment

4.1 Pull Requests

4.1.1 Features

- Implemented pipefail feature. If at least one command in pipeline returns an error, the whole pipeline returns an error.[18].
- Improved error message for not declared variable. Now shows the name of not found variable [15].
- Improved error message for invalid index. Message now shows the length of sequence where the out of bounds error occurs [14].
- Implemented work around to make file scheme work in Ion Shell inside Redox Os [19]
- Implemented subst method for Ion Shell. [9]

4.1.2 Bug fixes

- Make all buitlins of ion shell included the ion manual after building documentation [20].
- Ensured that spaces between left parentheses and the first argument does not cause an error anymore .[21]
- Converted unguarded unsafe conversion to utf-8 string. [22]

Related to redoxer

- Fixed semantic version of redoxer and applied missing argument for a function to fix compiler failure [23].

4.1.3 Code Quality

- Scripted option to only run one selected integration test. [24]
- Removed redundant, more than once, termination before tokenisation. [25]
- Remove obsolete parameter in several functions in the parser section of Ion Shell. [26]
- Fixing warnings by using modern API of criterion in benchmarks. [27]
- Added unit tests for parsing let statements. [28]
- Added unit tests for terminating lines before tokenisation. [29]
- Removed not used struct field. [30]

4.1.4 Documentation

- Moved msrv from clippy file to Cargo.toml. [31]
- Documented how the documentation of builtin are generated into the Ion Shell online manual. [32]
- Wrote section about how integration test work and how to write them. [33]
- Provided additional command for building ion manual locally. [34]
- Fixed formatting in the builtin section of the ion manual [35]
- Adjusted msrv to sync it within the project [36]
- Added notes/links to ion plugin. [37]
- Documented feature of initializing file. [38]
- Added output to example of creating an array withing the ion manual [39]
- Created a section about how integration tests work in Ion Shell. It also contains guides how to write integration tests for Ion Shell [13].

4.1.5 Configuration

- Make Linux job work in CI and fixed version in MakeFile. [40]
- Removed always overwritten test files and made it ignored in git. [41]
- Updated section about joining the redox community on matrix server. [42]

4.2 Issues

Resolved issues mean that they could be closed due to my contributions.

4.2.1 Opened and resolved Issues

- Reported problem of not working builtin documentation generation within ion manual [43]. Solved by pull request [20].
- Reported bug that white spaces between opening parentheses and the first argument causes an error [44]. Solved by pull request [44].
- Reported that ion manual is not updated by the documentation of the builtin methods [45]. Solved by pull request [46].
- Reported that the test suite of Ion Shell does not compile with intended rust version [47]. Solved by pull request [40].

4.2.2 Resolved Issues

The following issues were not opened by me but resolved by me via contribution.

- Feature request of new builtin method subst to allow for replacing empty arrays with an default array [11]. Resolved by pull request [9].

4.2.3 Opened issues

This issues were opened by me to report an problem. But they were not solved by me.

Related to redoxer

- Requested the rebuilding of the docker image for redoxer so that the Ion Shell repository can be build with a new rust version [48].

4.2.4 Still Opened Issues

This issues were opened by me to report an problem. But they have not been solved yet.

- Noted null-coalescing operator as a possible language feature [49]. Was brought up in the discussion in the issue [9].
- Noted down the trenary as a possible language feature [50]. Was brought up in the discussion in the issue [9].
- Reported the missing feature of handling redox os schemes in general [51].
- Proposed more information for error messages to end user. Espeacally by providing the locatoin where the error occured, ake line and colum [16].
- Reported general failing of Ci for ion shell [52].
- Reported that a job called "linux:stable" in the Ci still fails [53].

Related to redoxer.

- Reported that command "redoxer exec true" does fail on fedora [54].

List of Figures

List of Tables

List of Listings

Bibliography

[1] Redoxer Developers, "Gitlab ion shell respository," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion, (Accessed on: 2023-09-12).

[2] Ion Shell Developers, "Ion shell respository," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion, (Accessed on: 2023-09-12).

[3] ——, "Ion online manual," [Online]. Available at: https://doc.redox-os.org/ion-manual/, (Accessed on: 2023-09-12).

[4] ——, "Ion shell types," [Online]. Available at: https://doc.redox-os.org/ion-manual/variables/00-variables.html, (Accessed on: 2023-09-12).

[5] Florian Naumann, "String methods in ion shell," [Online]. Available at: https://doc.redox-os.org/ion-manual/expansions/06-stringmethods.html, (Accessed on: 2023-09-16).

[6] ——, "Arrays methods in ion shell," [Online]. Available at: https://doc.redox-os.org/ion-manual/expansions/07-arraymethods.html, (Accessed on: 2023-09-16).

[7] Rust Developers, "Testing private functions," [Online]. Available at: https://doc.rust-lang.org/book/ch11-03-test-organization.html#testing-private-functions, (Accessed on: 2023-09-12).

[8] Ion Shell Developers, "Ion shell contributing guide lines," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/blob/master/CONTRIBUTING.md?ref_type=heads, (Accessed on: 2023-09-12).

[9] Florian Naumann, "Implementation of subst method for arrays," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1238, (Accessed on: 2023-09-12).

[10] Nils, Eriksson matu3ba, Florian Naumann, "Issue parameter substitution on arrays," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1001, (Accessed on: 2023-09-12).

[11] Florian Naumann, "Parameter substitution on arrays," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1001, (Accessed on: 2023-09-12).

[12] ——, "Subst test code," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/blob/master/tests/subst.ion?ref_type=heads, (Accessed on: 2023-09-12).

[13] ——, "docs: added section on how to create integration tests and how they work," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1243, (Accessed on: 2023-09-16).

[14] ——, "feat: invalid index/range and length of sequence are provided for out of bounds error messsage to end user." [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1252, (Accessed on: 2023-09-12).

[15] ——, "feat: not found variable error now shows name of not found variable," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1253, (Accessed on: 2023-09-12).

[16] ——, "Enrich error messages information where the error occurred," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1022, (Accessed on: 2023-09-12).

[17] ——, "Build docker image of redoxer and publish new version on dockerhub," [Online]. Link to respository: https://gitlab.redox-os.org/redox-os/ion, (Local file path: src -> lib -> shell -> flow.rs).

[18] ——, "feat: pipefail option makes pipe return error code if any non zero," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1261, (Accessed on: 2023-09-12).

[19] ——, "fix(redox): file: prefix does not break expansion in ion shell anymore," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1239, (Accessed on: 2023-09-12).

[20] ——, "fix: man of history builtin is included into ion manual," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1247, (Accessed on: 2023-09-12).

[21] ——, "fix: spaces between 1. variable and opening parentheses, ignored for variable," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1250, (Accessed on: 2023-09-12).

[22] ——, "fix: in terminator removed unsafe by expect on converting bytes into utf8 str," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1261, (Accessed on: 2023-09-12).

[23] ——, "Bumped version of redox installer and fixed compile install errors," [Online]. Available at: https://gitlab.redox-os.org/redox-os/redoxer/-/merge_requests/9, (Accessed on: 2023-09-12).

[24] ——, "feat: single integration test is selectable," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1248, (Accessed on: 2023-09-12).

[25] ——, "fix: removed redundant termination of input string," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1255, (Accessed on: 2023-09-12).

[26] ——, "fix: Removed not needed parameter as builtin map for statement lexing," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1259, (Accessed on: 2023-09-12).

[27] ——, "fix: fixed warning by replacing deprecated criterion api," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1258, (Accessed on: 2023-09-12).

[28] ——, "fix: added unit test for let in grammar," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1257, (Accessed on: 2023-09-12).

[29] ——, "feat: added test for terminator and todos about strange edge cases," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1256, (Accessed on: 2023-09-12).

[30] ——, "fix: removed not used flag field," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1244, (Accessed on: 2023-09-12).

[31] ——, "fix: removed redundant termination of input string," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1254, (Accessed on: 2023-09-12).

[32] ——, "docs: note about builtin.md is generated by make manual," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1249, (Accessed on: 2023-09-12).

[33] ——, "docs: added section on how to create integration tests and how they work," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1243, (Accessed on: 2023-09-12).

[34] ——, "docs: adding commands to open manual in browser and removed not needed destination directory," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1245, (Accessed on: 2023-09-12).

[35] ——, "fix: ignores always overwritten doc builtin file and fixes wrong formatting at generating doc builtin file," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1246, (Accessed on: 2023-09-12).

[36] ——, "fix(version): adjusted other places for minimal used rust version to 1.56.0," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1241, (Accessed on: 2023-09-12).

[37] ——, "fix(version): adjusted other places for minimal used rust version to 1.56.0," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1240, (Accessed on: 2023-09-12).

[38] ——, "docs: describes feature for executing commands at start via initrc," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1234, (Accessed on: 2023-09-12).

[39] ——, "docs: Added missing echo output for create array in manual," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1233, (Accessed on: 2023-09-12).

[40] ——, "fix: partial ci fix and fix compiler error for make tests," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1237, (Accessed on: 2023-09-12).

[41] ——, "Pull request: removed always overwritten test file for git statging," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1236, (Accessed on: 2023-09-12).

[42] ——, "Pull request: removed always overwritten test file for git statging," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1235, (Accessed on: 2023-09-12).

[43] ——, "Locally generated builtin file for ion manual does not show a man page section for all builtins of ion." [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1029, (Accessed on: 2023-09-12).

[44] ——, "Spaces between opening parentheses of method and 1. argument raises error," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1021, (Accessed on: 2023-09-12).

[45] ——, "Bug: ion manual is not updated automatically after a commit," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1019, (Accessed on: 2023-09-12).

[46] ——, "Implementation of subst method for arrays," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/merge_requests/1238, (Accessed on: 2023-09-12).

[47] ——, "Make tests does not compile with indented toolchain 1.53.0," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1018, (Accessed on: 2023-09-12).

[48] ——, "Request of publishing new version redoxer on dockerhub," [Online]. Available at: https://gitlab.redox-os.org/redox-os/redoxer/-/issues/7, (Accessed on: 2023-09-12).

[49] ——, "null-coalescing operator," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1024, (Accessed on: 2023-09-12).

[50] ——, "ternary operator for arrays/strings," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1023, (Accessed on: 2023-09-12).

[51] ——, "ion cannot handle wildcards containing scheme prefixes like disk:," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1025, (Accessed on: 2023-09-12).

[52] ——, "Ci is broken with current base image redoxos/redoxer from docker hub." [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1020, (Accessed on: 2023-09-12).

[53] ——, "Job linux:stable fails with compile error in step make tests." [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1027, (Accessed on: 2023-09-12).

[54] ——, "Bug: Command redoxer exec true does not work on fedora," [Online]. Available at: https://gitlab.redox-os.org/redox-os/ion/-/issues/1022, (Accessed on: 2023-09-12).

[55] Redoxer Developers, "Gitlab redoxer respository," [Online]. Available at: https://gitlab.redox-os.org/redox-os/redoxer, (Accessed on: 2023-09-12).

Glossary

library A suite of reusable code inside of a programming language for software development. i

redoxer Rust program to execute another rust program within an environment resembling Redox. It is used in the CI of most projects in the redox os ecosystem. It is meant to check if additions to a code base, do not prevent programs from running in Redox. Link to repository of redoxer [55]. i, 15, 17, 18

shell Terminal of a Linux/Unix system for entering commands. i