

Text Data and Search Results

Urunk reimplementation

Group 20

Introduction

- What is Urank?
- Use case description
- User requirements
- Implementation
- User Interface

Urank tool

- What is urank?
- How do we use it?
- What is achieved with urank?



Use case description

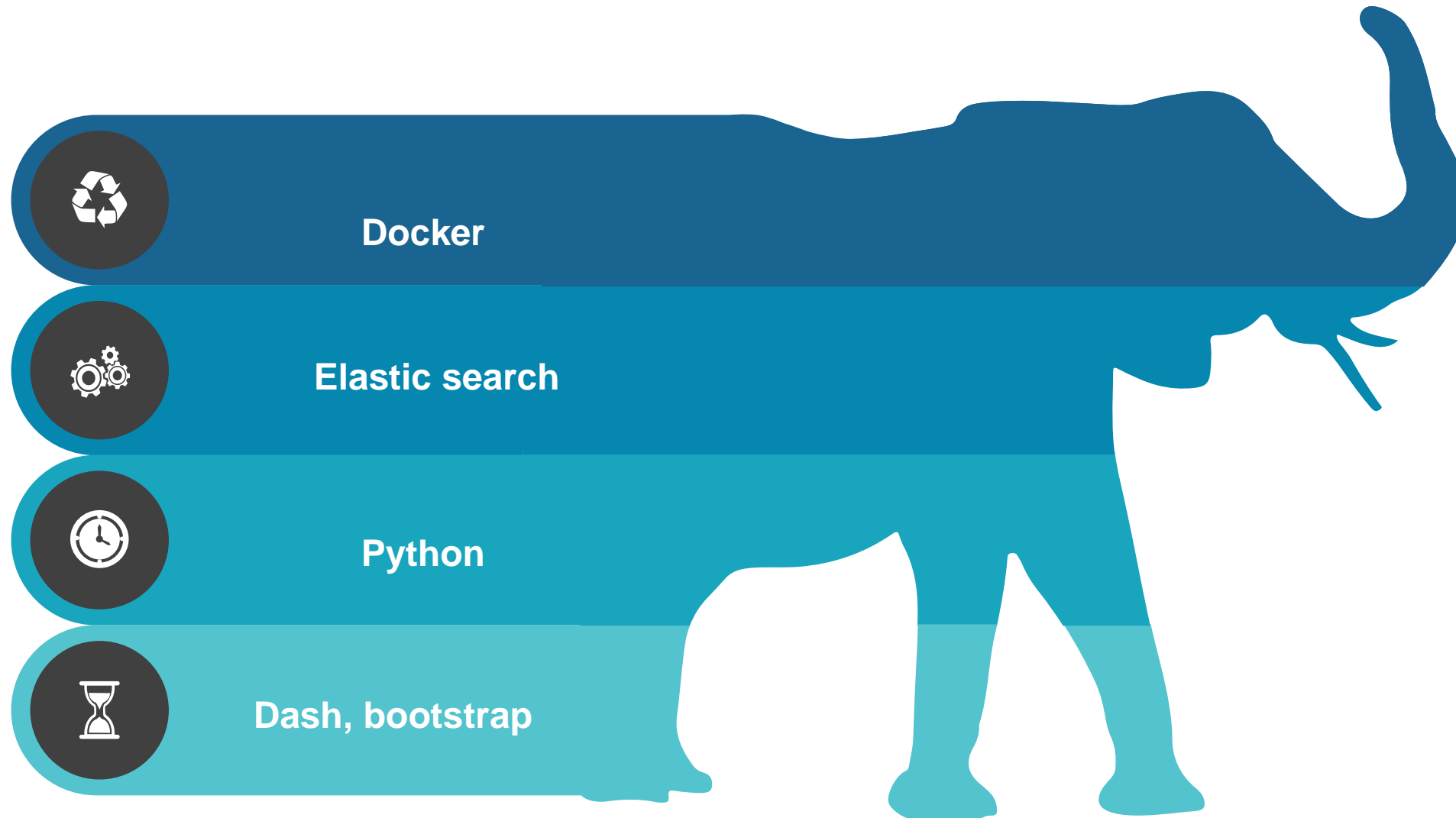
Our project is based on urank reimplementation where we are going to provide additional features which help users to get more details about information he is searching for

- Provide users with topics
- Search feature
- Inspect documents
- Inspect co-relation between them
- Bookmark
- History

User requirements

- Information search and retrieval
- Matching documents
- Ratio between documents and words
- Inspection of documents and searched words during inspection
- Memorize latest search results
- Easy access to the documents of interest

Implementation



Choose a topic

drivers

Type in words

valuable

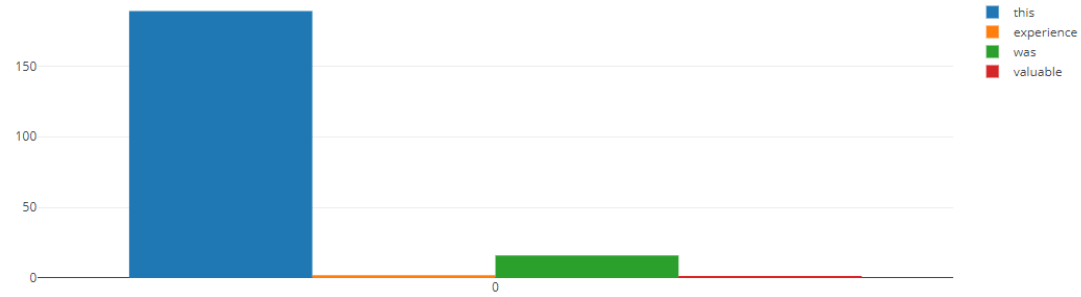
× this × experience × was
× valuable

Writing Linux Device Drivers in Assembly Language - 2002

Writing Network Drivers in Go (2018-ixy-go)

Writing Network Drivers in Rust (2018-ixy-rust)

Writing Linux Device Drivers in Assembly Language - 2002



View ★

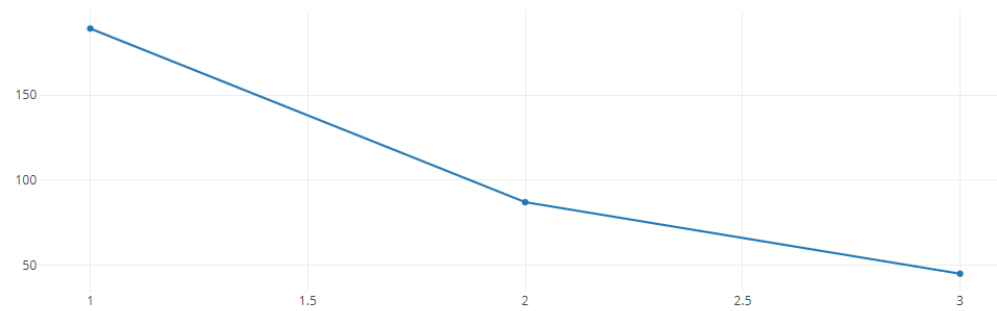
this

experience

was

valuable

this



Results

use in their examples) didn't seem particularly productive, either. What to do? It occurred to me that if I were to translate all the examples in C to a different programming language, I would have to really understand material. Of all the languages besides C, assembly is probably the most practical language with which one can write device drivers (practical from a capability point of view, as opposed to a software engineering point of view). The only reasonable language choice for Linux device drivers other than C is assembly language (that is, I *know* that I'd be able to write my drivers in assembly since GCC emits assembly code; I'm not sure at all that it would be possible to do **this** in some other language I have access to).

Rewriting Rubini & Corbet's examples in a different language certainly helps me understand what they're doing; rewriting their examples in assembly language really forces me to understand the concepts because C hides a lot of gory details from you (which Rubini & Corbet generally don't bother to explain). So that **was** my second reason for using assembly; by using assembly to write these drivers I *really* have to know what's going on. **This experience was valuable** not only because it forced me to learn Linux device drivers to a depth I would have never otherwise attained, but it also taught me a lot of in-depth *Linux* programming, as well. You won't fully appreciate the complexity of the Linux system until you've converted a large number of C header files to assembly language (and verified that the conversion is correct).

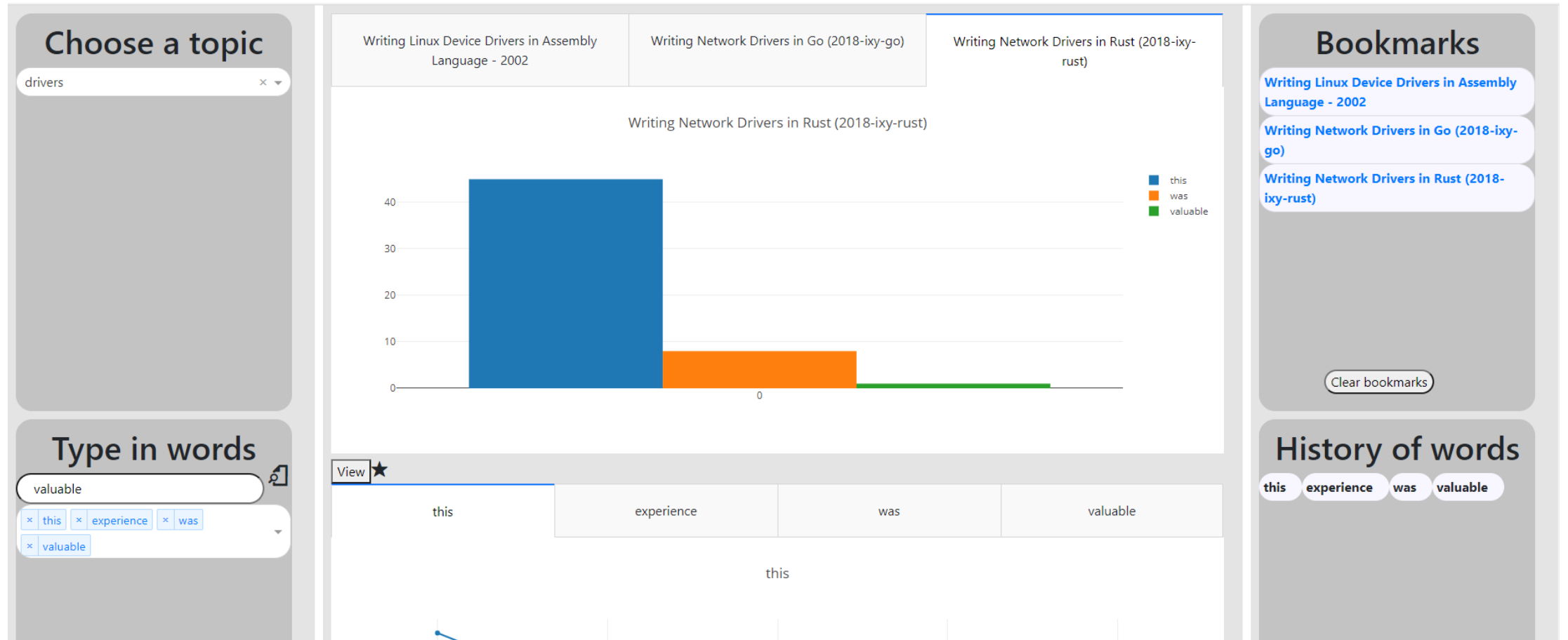
Note that all the examples in **this** text are pure assembly language. I don't write a major portion of the driver in C and then call some small assembly language function to handle some operation. That would defeat the purpose for (my) using assembly language in the first place, that is, forcing me to really learn **this** stuff.

Of course, many people really want to know how to write Linux device drivers in assembly language. Either they prefer assembly over C (and many people do, believe it or not), or they need the efficiency or device control capabilities that only assembly language provides. Such individuals will probably find **this** document enlightening. While those wanting more efficient code or more capability could probably use the C+assembly approach, they should still find **this** document interesting.

Of course, any die-hard Unix/Linux fan is probably screaming "don't, don't, don't" at **this** point. "Why on Earth would anyone be crazy enough to write a document about assembly language drivers for Linux?" they're probably saying. "Doesn't **this** fool (me) know that Linux runs on different platforms and assembly drivers won't be portable?" Of course I realize **this**. I'm also assuming that anyone bright enough to write a

Highlighted PDF

Bookmarks





THANK YOU

