Charles Bennington
4/15/2025

Linux and Devops Write Up

Website Link : https://cbennington852.github.io/MyPL-a-programming-language/

NOTE: The makefile and other related files are in the “packaging” folder

○ **A list of techniques and tools used to convert the project.**

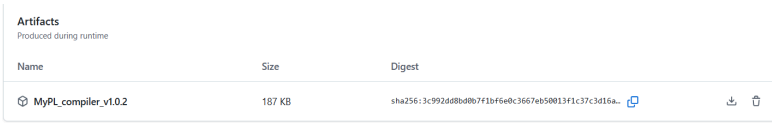
1. *GitHub Pages:* I used a free tool called “github pages”, which allows you to render a static .html page from their website. I used this tool to make the “downloads page” for the compiler project.
   https://cbennington852.github.io/MyPL-a-programming-language/
2. *dpkg:* I used dpkg to make a .deb file for the compiler to be installed in.
3. *Linux Alias*: I made use of a linux alias to make the debian project callable form the command line using only the command
   mypl example.mypl
4. OPL Final project: I used my OPL final project, which is a class offered at gonzaga, where you write your own programming language. The language is called mypl. You can find examples of mypl in the examples folder.

○ **A description of key steps taken to build the release package (e.g., Debian package, Docker image).**

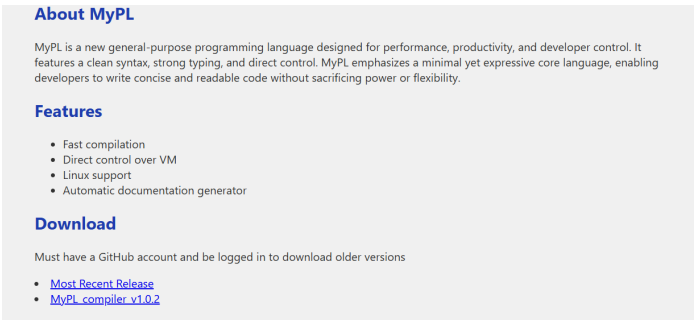
1. The compiler is wrapped up into a nice little debian package.

```
- name: build debian file
  run: cd packaging/ && make build-deb
```

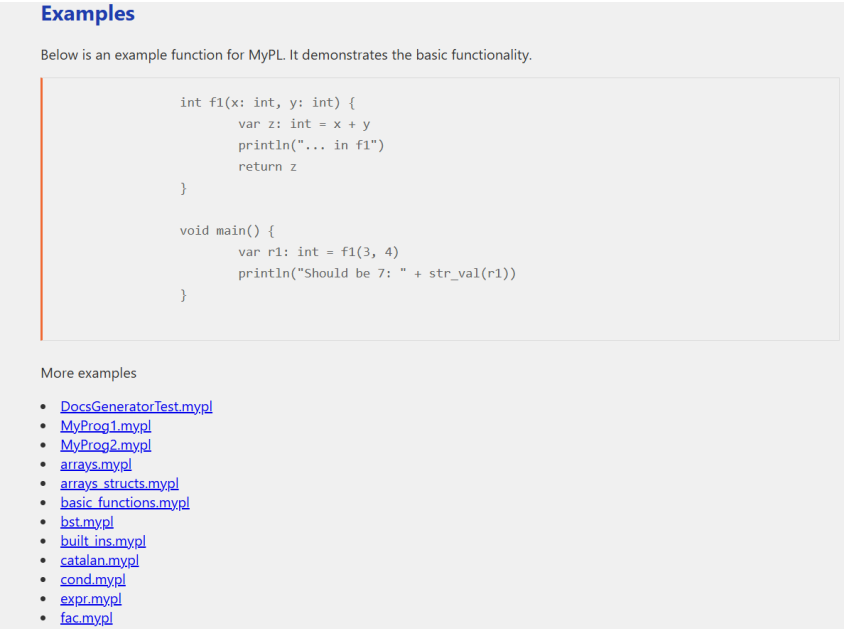
2. That debian package is saved as a github artifact on a GitHub tags push.



3. The compiler download page (accessible via this link) then adds a new section to its “current download section” allowing users to decide which version to download. The Most recent release links to the github main repo.



4. The program examples are also available from the webpage, The website is directly pulled from the GitHub website using the GitHub API. Each link should work.



Examples

Below is an example function for MyPL. It demonstrates the basic functionality.

```
int f1(x: int, y: int) {
	var z: int = x + y
	println("... in f1")
	return z
}

void main() {
	var r1: int = f1(3, 4)
	println("Should be 7: " + str_val(r1))
}
```

More examples

- DocsGeneratorTest.mypl
- MyProg1.mypl
- MyProg2.mypl
- arrays.mypl
- arrays_structs.mypl
- basic_functions.mypl
- bst.mypl
- built_ins.mypl
- catalan.mypl
- cond.mypl
- expr.mypl
- fac.mypl

○ **A discussion of challenges encountered during the project.**
The most difficult part of this was figuring out how alias's work in linux, I wanted the compiler to be able to be callable via "mypl example.mypl" instead of the user having to type out the path to the .jar file.

There are two separate files to store alias's in. One that is user specific, and one that is computer specific, I ended up having the alias installed on the computer specific one.

○ **An evaluation of how DevOps methodologies can benefit your future projects.**
In the future I will strive to automate things, and make deployment a more painless process than it has been in the past.

One of my favorite things about this class has been getting to spend time on the linux terminal. I think that learning more about Linux will be good for my career in the future. In general learning how to script things makes it a lot easier, because it means that I can automate things to make my life easier.

○ **A reflection on how this Linux & DevOps course has influenced your perspective on software development and related technologies.**

This course has made me realise that software is an integrated process, and that deployment should be the first thing that you set up when making a new project. I have enjoyed the information about docker containers and the kubernetes section. An excuse to spend more time on linux systems is also greatly appreciated.