Milestone 2

Task Matrix

Progress of current Milestone:

Task	Completion	Ian	Dylan	Todo
Basic Qemu Image for Compiler Theory	100%	1%	99%	Experiment with QEMU smb
Run Commands and Provide/Receive Standard Output/Input	90%	70%	20%	Fix current bugs with reading from standard input
Import and Export Files from an Image	100%	99%	1%	

Compiler Theory Container

Setting up a container for the Compiler Theory course

Requirements

GNU

Must have GNU Toolchain and Binutils

Malloc

Must be able to allocate memory using malloc

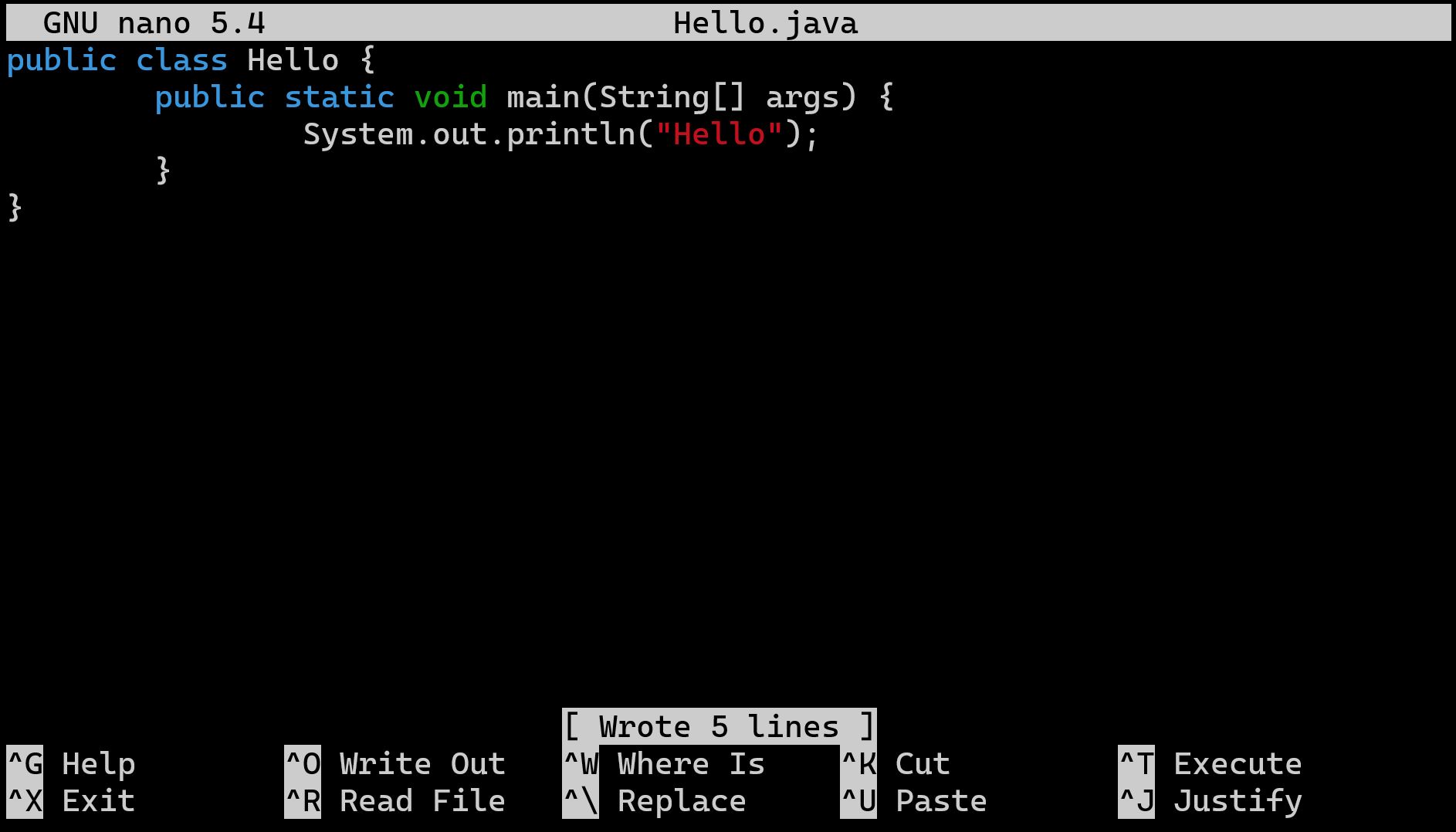
Java

Must be able to compile and run Java programs

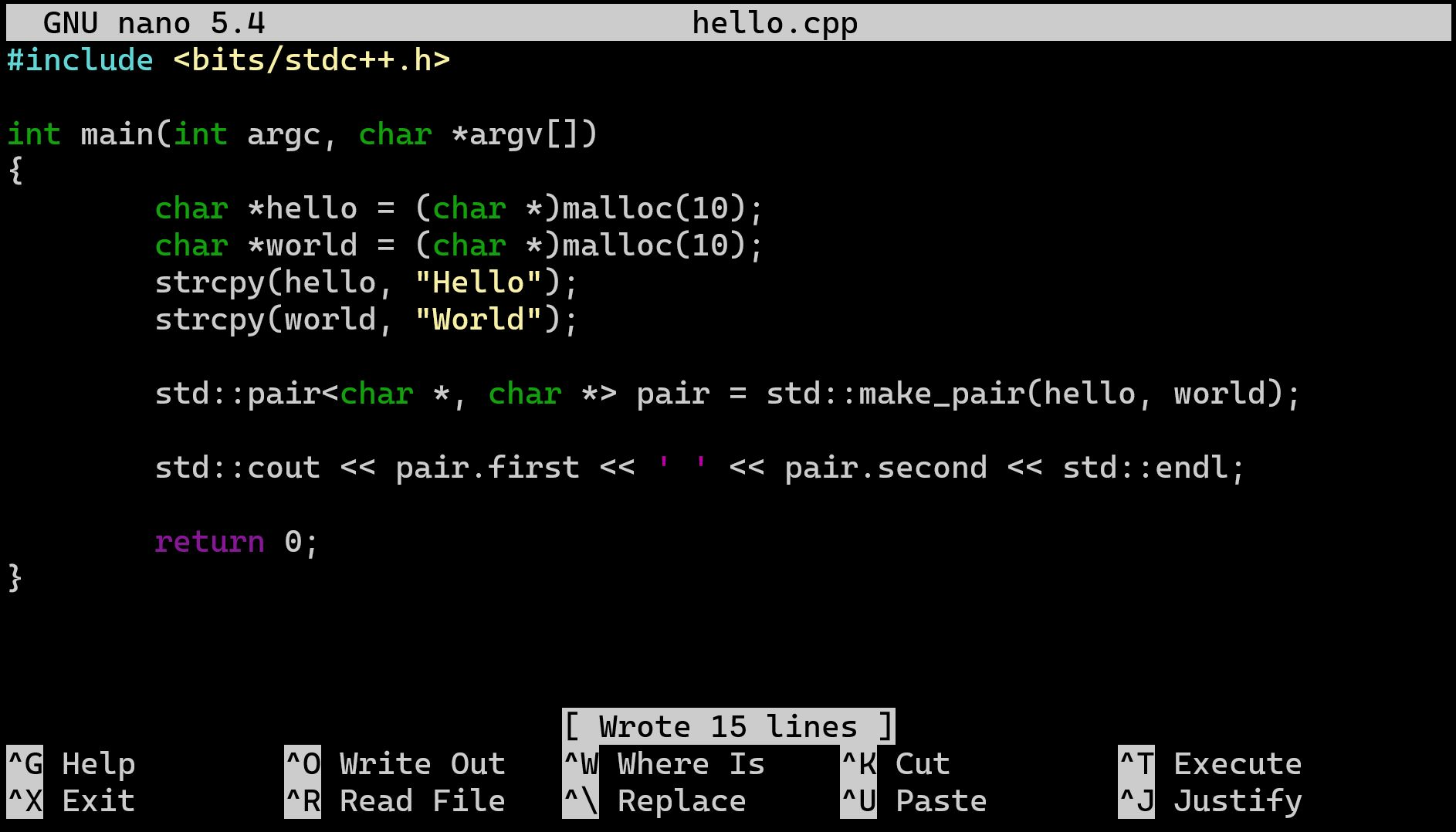
SPARC

Must be able to run SPARC binaries

```
root@debian:~/presentation# nano Hello.java
```


GNU nano 5.4 Hello.java
public class Hello {
public static void main(String[] args) {
System.out.println("Hello");
}
}
[Wrote 5 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

```
root@debian:~/presentation# javac Hello.java
root@debian:~/presentation# java Hello
Hello
root@debian:~/presentation#
```


GNU nano 5.4 hello.cpp
#include <bits/stdc++.h>
int main(int argc, char *argv[])
{
char *hello = (char *)malloc(10);
char *world = (char *)malloc(10);
strcpy(hello, "Hello");
strcpy(world, "World");
std::pair<char *, char *> pair = std::make_pair(hello, world);
std::cout << pair.first << ' ' << pair.second << std::endl;
return 0;
}
[Wrote 15 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

```
root@debian:~/presentation# sparc-linux-g++ hello.cpp
root@debian:~/presentation#
```

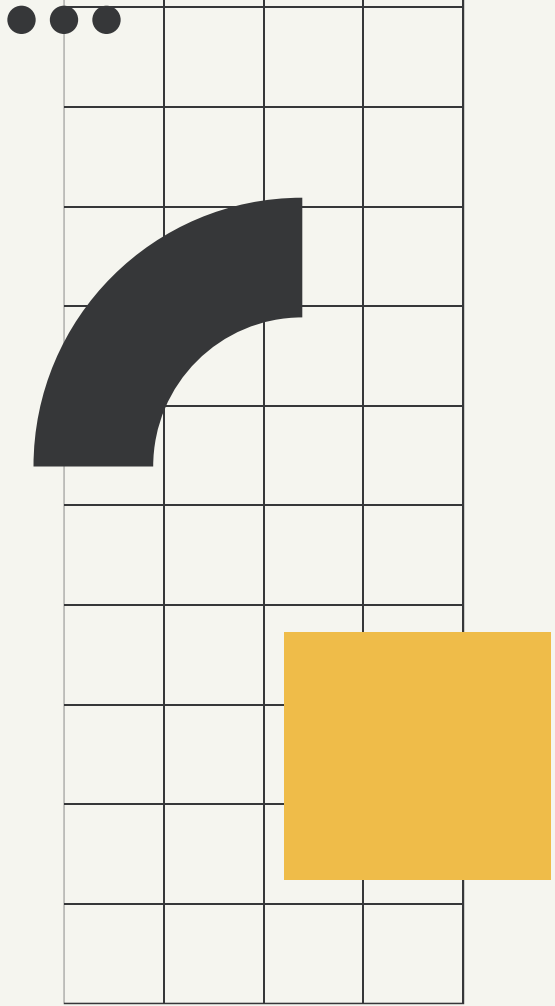
```
root@debian:~/presentation# sparc-linux-g++ hello.cpp
root@debian:~/presentation# file a.out
a.out: ELF 32-bit MSB pie executable, SPARC, version 1 (SYSV), dynamically lin
ked, interpreter /lib/ld-uClibc.so.0, not stripped
root@debian:~/presentation#
```

```
root@debian:~/presentation# sparc-linux-g++ hello.cpp
root@debian:~/presentation# file a.out
a.out: ELF 32-bit MSB pie executable, SPARC, version 1 (SYSV), dynamically lin
ked, interpreter /lib/ld-uClibc.so.0, not stripped
root@debian:~/presentation# ./a.out
Hello World
root@debian:~/presentation#
```

```
root@debian:~/presentation# sparc-linux-g++ hello.cpp
root@debian:~/presentation# file a.out
a.out: ELF 32-bit MSB pie executable, SPARC, version 1 (SYSV), dynamically lin
ked, interpreter /lib/ld-uClibc.so.0, not stripped
root@debian:~/presentation# ./a.out
Hello World
root@debian:~/presentation# mipsel-linux-g++ hello.cpp
root@debian:~/presentation#
```

```
root@debian:~/presentation# sparc-linux-g++ hello.cpp
root@debian:~/presentation# file a.out
a.out: ELF 32-bit MSB pie executable, SPARC, version 1 (SYSV), dynamically lin
ked, interpreter /lib/ld-uClibc.so.0, not stripped
root@debian:~/presentation# ./a.out
Hello World
root@debian:~/presentation# mipsel-linux-g++ hello.cpp
root@debian:~/presentation# file a.out
a.out: ELF 32-bit LSB pie executable, MIPS, MIPS32 version 1 (SYSV), dynamical
ly linked, interpreter /lib/ld-uClibc.so.0, not stripped
root@debian:~/presentation#
```

```
root@debian:~/presentation# sparc-linux-g++ hello.cpp
root@debian:~/presentation# file a.out
a.out: ELF 32-bit MSB pie executable, SPARC, version 1 (SYSV), dynamically lin
ked, interpreter /lib/ld-uClibc.so.0, not stripped
root@debian:~/presentation# ./a.out
Hello World
root@debian:~/presentation# mipsel-linux-g++ hello.cpp
root@debian:~/presentation# file a.out
a.out: ELF 32-bit LSB pie executable, MIPS, MIPS32 version 1 (SYSV), dynamical
ly linked, interpreter /lib/ld-uClibc.so.0, not stripped
root@debian:~/presentation# ./a.out
Hello World
root@debian:~/presentation#
```



Container Manager

Python program that manages the containers on the user's system

Functionality

Start Containers

Containers can be started with an open SSH connection

Send Commands

Containers can be sent commands that allows users to interact with the process's IO

Get/Put Files

Containers can be sent files, and containers can have files extracted

Stop Containers

Containers can be stopped, freeing all of the resources they were using

Demo

```
PS C:\Users\iworz\Documents\jabberwocky-container-manager> poetry run python run.py
INFO:__main__:Executing "C:\Program Files\qemu\qemu-system-x86_64" -monitor null -net nic -net user,host
fwd=tcp::12300-:22 -serial stdio -nographic -m 500M -drive file=hdd.qcow2,format=qcow2
INFO:__main__:Executing ssh -oStrictHostKeyChecking=no -oLogLevel=ERROR -oPasswordAuthentication=no -i C
:\Users\iworz\.containers\ct\id_rsa -p 12300 root@localhost ls
a.out
echo.c
INFO:__main__:Executing ssh -oStrictHostKeyChecking=no -oLogLevel=ERROR -oPasswordAuthentication=no -i C
:\Users\iworz\.containers\ct\id_rsa -p 12300 root@localhost rm -f echo.c
INFO:__main__:Attempting put(echo.c, echo.c)
INFO:__main__:Executing ssh -oStrictHostKeyChecking=no -oLogLevel=ERROR -oPasswordAuthentication=no -i C
:\Users\iworz\.containers\ct\id_rsa -p 12300 root@localhost sparc-linux-gcc echo.c
INFO:__main__:Executing ssh -oStrictHostKeyChecking=no -oLogLevel=ERROR -oPasswordAuthentication=no -i C
:\Users\iworz\.containers\ct\id_rsa -p 12300 root@localhost ./a.out
Enter string: (Under 100 characters please): Hello
ECHO: Hello
INFO:__main__:Attempting get(a.out, a.out)
INFO:__main__:Attempting to poweroff
INFO:__main__:Success!
PS C:\Users\iworz\Documents\jabberwocky-container-manager>
```

Next Milestone Goals

Command-line interface

- We want to create a command-line interface for end-users to be able to manage containers, execute command inside of them, access their internal shell, and other actions listed in the design document.
- This will be written in Python

CLI Shell Scripts

- The end user should not have to care that the CLI is written in Python
- We want to create a set of shell scripts which will indirectly call our program with the appropriate Python completely seamlessly.

Installer

- We would like the end user to be able to install our program and its dependencies without much fuss.
- For this we will need to create different installers for all of our supported platforms.
  - Windows
  - macOS
  - Ubuntu
  - Debian

Task	Ian	Dylan
Implement, demo, and test command-line interface support in Python	99%	1%
Implement, demo, and test command-line interface as shell and batch scripts to send information to Python	50%	50%
Implement, demo, and test installer or installation guide for installing system on Windows, MacOS, and Debian	1%	99%

Questions?