

NAME: Ben Puryear

FILE: hw1-writeup.pdf

DATE: Spring 2023

DESC: This pdf will provide a short description of any challenges and/or issues I faced and how I addressed them in this assignment.

### Challenges faced:

The only challenge I faced during this assignment were figuring out how to stop reading command line input after Ctrl-d is pressed without stopping the execution of the entire program. I overcame this challenge by coming to the realization that I can use &cin as if it was a file and check if it has reached EOF().

### Testing:

#### Basic From-File usage:

```

bpuryear@ada:~/hw1-Ben10164$ ./mypl --help
Usage: ./mypl [option] [script-file]
Options:
    --help    prints this message
    --lex     displays token information
    --parse   checks for syntax errors
    --print   pretty prints program
    --check   statically checks program
    --ir      print intermediate (code) representation
bpuryear@ada:~/hw1-Ben10164$ ./mypl --lex examples/example.txt
[Lex Mode]
T
bpuryear@ada:~/hw1-Ben10164$ ./mypl --parse examples/example.txt
[Parse Mode]
Th
bpuryear@ada:~/hw1-Ben10164$ ./mypl --print examples/example.txt
[Print Mode]
There
bpuryear@ada:~/hw1-Ben10164$ ./mypl --check examples/example.txt
[Check Mode]
There sat that beautiful big machine whose sole job was to copy things
bpuryear@ada:~/hw1-Ben10164$ ./mypl --ir examples/example.txt
[IR Mode]
There sat that beautiful big machine whose sole job was to copy things
and do addition. Why not make the computer do it? That's why I sat
down and wrote the first compiler. It was very stupid. What I did was
watch myself put together a program and make the computer do what I
did. -Grace Hopper
bpuryear@ada:~/hw1-Ben10164$

```

#### Invalid Usage:

```

bpuryear@ada:~/hw1-Ben10164$ ./mypl --fix
ERROR: Unable to open file '--fix'
bpuryear@ada:~/hw1-Ben10164$ ./mypl examples/nosuchfile.txt
ERROR: Unable to open file 'examples/nosuchfile.txt'
bpuryear@ada:~/hw1-Ben10164$ ./mypl a b c
Usage: ./mypl [option] [script-file]
Options:
    --help    prints this message
    --lex     displays token information
    --parse   checks for syntax errors
    --print   pretty prints program
    --check   statically checks program
    --ir      print intermediate (code) representation
bpuryear@ada:~/hw1-Ben10164$ >

```

Basic From-CLI usage:

```

bपुर्यार@ada:~/hw1-Ben10164$ ./mypl --lex
[Lex Mode]
foo bar baz
f
bपुर्यार@ada:~/hw1-Ben10164$ ./mypl --parse
[Parse Mode]
foo bar baz
fo
bपुर्यार@ada:~/hw1-Ben10164$ ./mypl --print
[Print Mode]
foo bar baz
foo
bपुर्यार@ada:~/hw1-Ben10164$ ./mypl --check
[Check Mode]
foo bar baz
foo bar baz
bपुर्यार@ada:~/hw1-Ben10164$ ./mypl --ir
[IR Mode]
foo bar baz
qux quux quuz corge
foo bar baz
qux quux quuz corge
bपुर्यार@ada:~/hw1-Ben10164$ ./mypl
[Normal Mode]
foo foo foo
bar bar bar
baz baz baz
foo foo foo
bar bar bar
baz baz baz
bपुर्यार@ada:~/hw1-Ben10164$

```

My own text example (Lorem Ipsum Generator)

```

bपुर्यार@ada:~/hw1-Ben10164$ ./mypl examples/myExaple.txt
[Normal Mode]
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Consectetur a erat nam at lectus u
rna dui convallis. Lectus sit amet est placerat. Purus sit amet luctus venenatis lectus magna fringilla urna porttitor. At risus viverra adipiscing at in. In
teger vitae justo eget magna. Semper eget dui at tellus at urna condimentum mattis pellentesque. Vivamus at augue eget arcu dictum. Ut sem nulla pharetra dia
m sit amet nisl suscipit. Scelerisque varius morbi enim nunc faucibus.

Congue eu consequat ac felis. At tempor commodo ullamcorper a. Sed augue lacus viverra vitae. Eget dolor morbi non arcu risus. Orci a scelerisque purus semper
eget dui at. Posuere urna nec tincidunt praesent semper feugiat. Ipsum dolor sit amet consectetur adipiscing elit. Nec dui nunc mattis enim ut. Nunc sed bla
ndit libero volutpat sed cras. Lectus urna dui convallis convallis tellus id interdum. In pellentesque massa placerat dui ultricies lacus sed turpis. Tincid
unt ornare massa eget egestas purus viverra accumsan in nisl. Nibh tellus molestie nunc non. Odio euismod lacinia at quis risus sed vulputate odio. Dis partur
ient montes nascetur ridiculus mus mauris vitae ultricies. In hac habitasse platea dictumst. Eget nunc scelerisque viverra mauris in aliquam sem fringilla ut.
Feugiat in ante metus dictum at. Sagittis id consectetur purus ut faucibus pulvinar elementum integer enim. Mauris commodo quis imperdiet massa tincidunt nun
c pulvinar sapien et.

bपुर्यार@ada:~/hw1-Ben10164$

```

Valgrind output:

```

bपुर्यार@ada:~/hw1-Ben10164$ valgrind ./mypl examples/example.txt
==384211== Memcheck, a memory error detector
==384211== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==384211== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==384211== Command: ./mypl examples/example.txt
==384211==
[Normal Mode]
There sat that beautiful big machine whose sole job was to copy things
and do addition. Why not make the computer do it? That's why I sat
down and wrote the first compiler. It was very stupid. What I did was
watch myself put together a program and make the computer do what I
did. -Grace Hopper

==384211==
==384211== HEAP SUMMARY:
==384211==   in use at exit: 0 bytes in 0 blocks
==384211==   total heap usage: 14 allocs, 14 frees, 83,941 bytes allocated
==384211==
==384211== All heap blocks were freed -- no leaks are possible
==384211==
==384211== For lists of detected and suppressed errors, rerun with: -s
==384211== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
bपुर्यार@ada:~/hw1-Ben10164$

```