# GDB.

The GNU Debugger.

"Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.
"

- Brian Kernighan

# Why is Debugging Important?

- Correctness
- Security
- Performance
- Cost-Efficiency
- and many more...

# Types of Errors

- Compile-time Errors
  - Syntax Errors
  - Type Mismatches
- Runtime Errors
  - Division-by-zero
  - Invalid memory access

# What is GDB?

**GCC - GNU Compiler Collection**

Collection of compilers for various programming languages.

Compiles source code into executable binaries

It detects and reports compile-time errors.

# What is GDB?

**GDB - The GNU Debugger**

Debugger for debugging executable binary files compiled using GCC.

Helps in finding and analyzing runtime errors and issues in the code.

# Using GDB

Compile the souce code using **gcc** with the **-g** flag.

```
gcc -g main.c -o main.out
```

GDB can also be used with **g++.** Once this is done, you can start the debugger by running,

```
gdb ./main.out
```

# Basic GDB Commands

# Setting Breakpoints

```
break main
break
break 42
break MyFunction
break MyFile:42
```

# Manipulating Breakpoints

```
delete 42
info breakpoints
```

# Running and Debugging

```
run
continue
finish
step
step 42
next
```

# Stack Trace and Variables

```
print var
backtrace
quit
help all
```

# Example on using GDB

# Using GDB for xv6

# Debugging xv6 User Programs

```
make qemu-gdb

gdb kernel/kernel

(gdb) target remote localhost:26000

(gdb) file user_exec
```

# Why do we need remote debugging for xv6?

- GDB runs separately from the program being debugged.
- They communicate over a simple medium such as a network socket.
- QEMU provides a remote debugging stub which GDB uses to debug the program
- This way user programs can be debugged while the xv6 kernel is running so that we can use the symbols populated by the kernel into the symtab.

# Thank You

GDB.