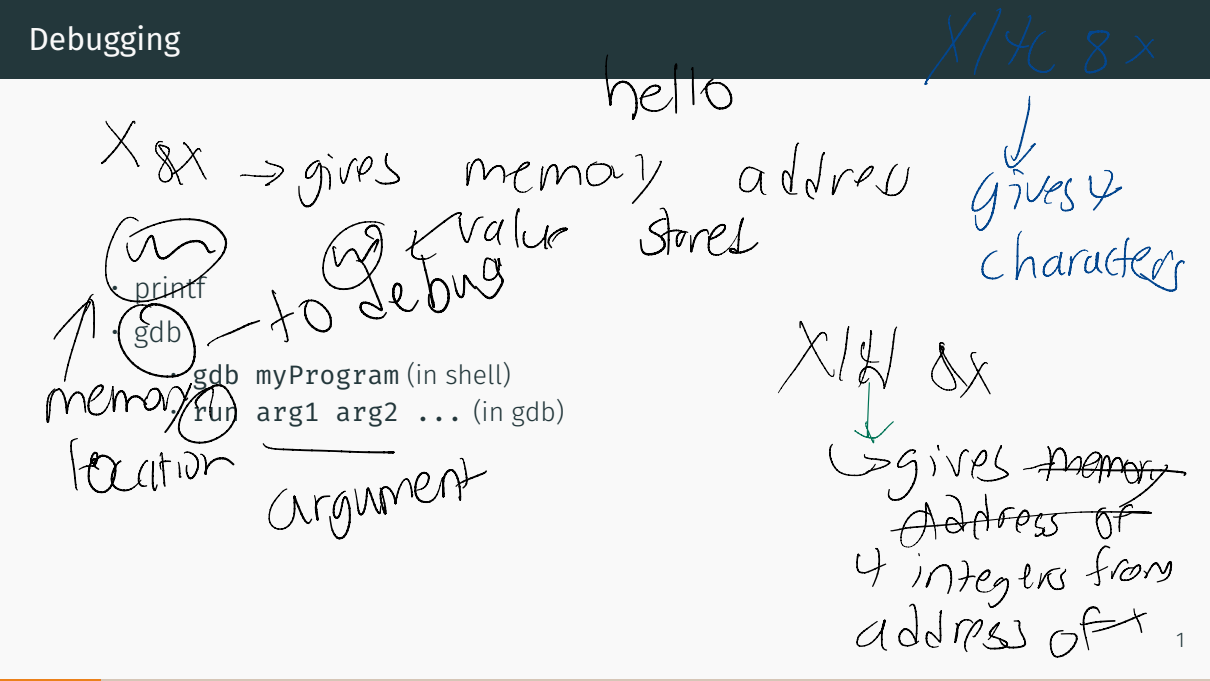
C debugging, building, etc.

CS 211: Computer Architecture

Fall 2020

Debugging


hello
X 8x → gives memory address
value stored
printf
gdb
to debug
gdb myProgram (in shell)
run arg1 arg2 ... (in gdb)
memory location
argument
X/4c 8x
gives 4 characters
X/4d 8x
gives memory address of 4 integers from address of

GDB

`break`	set a breakpoint
`run`	run program
`list`	show original source code
`step`	step to next line (into a function)
`next`	step to next line (over function calls)
`continue`	continue running after stopping
`kill`	kill program being debugged
`quit`	exit gdb and kill program
`print`	evaluate source expression
`x`	display memory contents
`bt`	show call stack
`frame`	select stack frame

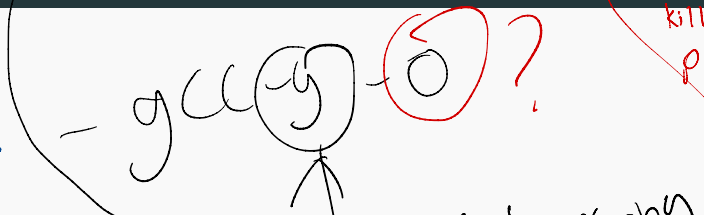
Makefiles

frame 2 ~ goes to line 2 of stack

frame 0 - first line of stack

Version 1:

```
hello: hello.c
	gcc -o hello hello.c
```



quit - quits gdb

kill - kills the program

backtrace - numbered from 0...n
↳ gives the whole callstack

- list gives lines before and after
- print x (at the current time)

- cat Makefile

  shows make file

- make

  gives command

Makefiles

Version 2:

```
hello: hello.c
    gcc -o $@ $<
```

- $@: target file name
- $<: first prerequisite
- $^: all prerequisites

https://www.gnu.org/software/make/manual/html_node/Automatic-Variables.html

Makefiles

Version 3:

```
all: hello

%: %.c
	gcc -o $@ $^
```

build any program that can accept any programs

make and make all same

Makefiles

Version 4:

```
OUTPUT=hello

all: $(OUTPUT)

clean:
	rm -f *.o $(OUTPUT)

%: %.c
	gcc -o $@ $^
```

gets rid of compile outfile

Version control with git

- git log
- git log --oneline
- rm delete file
- rm * brings it back
- git checkout

```
• git init
• git add *
• git commit -m "Initial files"
• (edit)
• git commit -m "add feature X"
```

int a[10];
a[10] = 20;
stack buffer overflow

heap buffer overflow — ex a[20] when made an array of size 20
heap make memory/init

Finding memory leaks (part memory error)

gcc -g -fsanitize=address -o memleak memleak.c
file name

Valgrind

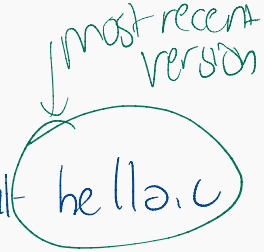
Sometimes Valgrind doesn't find memory leak

gcc -g -fsanitize=address -o memleak memleak.c
file name

Valgrind --leak-check=full ./memleak

git checkout

- git log
- git checkout
- git push



git check out
- takes you to previous change good

in order to go back
git checkout master

Memory leaks

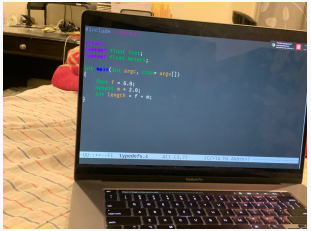
```
int main()
{
    int* p = malloc(20 * sizeof(int));
}
```

Typedef

create a new type but it is still a float

```
typedef float feet;
typedef float meters;

feet f = 6.0;
meters m = 2.0;
int length = f + m;
```



typedef struct Foo{
    int x;
    int y;
} Foo;

struct Foo* otherFoo;

use this struct it has not been declared yet

Typedef

```
typedef struct Foo
{
    // ...
    struct Foo* foo;
} Foo;
```

C preprocessor

#include "foo.h"

↓ make a variable

#define MAXSIZE 10

```
#define
#if
#ifdef
#ifndef
#include
```

#ifdef — if defined

#ifndef — if not defined

#pragma

squared(a+b)

a+b*a+b

but

a+(b*a)+b

So lets say we define two same structures