C debugging, building, etc.

CS 211: Computer Architecture

Fall 2020

Debugging

- printf
- gdb
  - `gdb myProgram` (in shell)
  - `run arg1 arg2 ...` (in gdb)

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

Version 1:

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

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 $@ $^
```

Version 4:

```
OUTPUT=hello

all: $(OUTPUT)

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

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

Version control with git

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

Version control with git

- `git log`
- `git checkout`
- `git push`

Memory leaks

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

Typedef

```
typedef float feet;
typedef float meters;

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

Typedef

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

C preprocessor

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