

# Hello Code Review

Tanner Johnson

# Contents

- Do not have access to video so this powerpoint is the submission along with hello.c and the makefile
- Go over hello.c code
- Look at output
- Explain the 3 system calls used by the program

# Code: Includes

- Shown here:

```
#include <pthread.h>    // pthread_create, pthread_join
#include <syslog.h>      // syslog()
```

- Pthread gives us access to the pthread\_create and pthread\_join system calls
- And in general gives access to the pthread POSIX library on linux

# Code: Hello Thread

- Shown here: 

```
void *helloThread(void *threadp)
{
    // Log the thread message at Critical debug level
    syslog(LOG_CRIT, "[COURSE:%u][ASSIGNMENT:%u] Hello World from Thread!", 1,1 );
}
```
- The thread of execution that is spawned from main.
- Simply uses syslog to log a hello message
- Will go over syslog() system call in greater detail later

# Code: Main

- Shown here:

```
int main (int argc, char *argv[])
{
    // Log the main hello message at Critical debug level
    syslog(LOG_CRIT, "[COURSE:%u][ASSIGNMENT:%u] Hello World from Main!\n", 1, 1);

    // Create and spawn a thread using the default thread attributes and whose
    // execution starts at the helloThread function. Pass no parameters to the
    // thread.
    pthread_t thread;
    pthread_create(&thread, NULL, helloThread, NULL);

    // Block until the helloThread has exited and do not store the return code
    pthread_join(thread, NULL);
}
```

- Logs hello from main
- Spawns a thread and waits till it exits for the program to exit
- Again will cover these system calls in depth in following slides

# System Call: syslog()

- User space API contained in <syslog.h>
- Man page: <https://man7.org/linux/man-pages/man3/syslog.3.html>
- Takes in a log level and a formatted C string
- This string and its level are passed to the syslogd linux daemon
  - Run `ps -aux | grep syslogd` to see its process
- The syslogd service logs the message in the appropriate log file and multiplexes multiple programs trying to log concurrently
- To see output of hello, cat the contents of `/var/log/syslog`

# Syscall: pthread\_create()

- Header found in pthread user api <pthread.h>
- Man page : [https://man7.org/linux/man-pages/man3/pthread\\_create.3.html](https://man7.org/linux/man-pages/man3/pthread_create.3.html)
- Creates and spawns a pthread
- First arg) pointer to pthread\_t to be populated with created thread attributes
  - To be used by spawning code to monitor and control thread
- Second arg) thread attributes including the scheduler to be used, priority, signal mask, etc
- Third arg) function pointer of function to start execution of thread
- Fourth arg) pointer to data to pass as parameter to thread.

# Syscall: pthread\_join()

- Header found in pthread user api <pthread.h>
- Man Page: [https://man7.org/linux/man-pages/man3/pthread\\_join.3.html](https://man7.org/linux/man-pages/man3/pthread_join.3.html)
- Blocks the caller until passed thread exits
- 1st arg) the pthread\_t meta data populated by a create call
- 2nd arg) pointer to pointer of int type. Populates a pointer that itself points to the return code / status of the thread.



# Output

```
hw3> make hello
gcc -O0 -g -c hello.c
gcc -O0 -g -o hello hello.o -lpthread
hw3> ./hello
hw3> tail -n 2 /var/log/syslog
Jan 14 14:30:47 tj-HP-Pavilion-Desktop-PC-570-p0XX hello: [COURSE:1][ASSIGNMENT:1] Hello World from Main!
Jan 14 14:30:47 tj-HP-Pavilion-Desktop-PC-570-p0XX hello: [COURSE:1][ASSIGNMENT:1] Hello World from Thread!
hw3> █
```