COMP 530H
Operating Systems Honors:

A Study of Linux on Intel IA-32 Systems

Introduction
Kevin Jeffay
Don Smith

What to Expect from Honors Course

◆ Introduction to Linux OS kernel implementation.
◆ Experience writing, testing, and evaluating code running in the Linux kernel.
◆ Integrated with the regular COMP 530 course:
  ✦ Same in-class material (Monday and Wednesday).
  ✦ Extra meeting Fridays for Linux material.
  ✦ Same problem sets and exams.
  ✦ Different programming assignments AFTER assignments 1 & 2.
    • Somewhat compressed schedule for turning in assignments 1 & 2.
◆ Key Difference – 530 students will use services provided in the Linux kernel; 530H students will develop new services for the Linux kernel.
Necessary Background

◆ C/C++ and Linux experience beyond COMP 411
  ✦ Heavy use of pointers, structures, and strings

◆ Experience with tools and utilities
  ✦ Any Linux editor (vim, vi, emacs, ….)
  ✦ gcc and make
  ✦ less, grep, cut, ….

Why This is Not for the Fainthearted

◆ No IDE (e.g., no Eclipse, MS Visual Studio, etc.)
◆ No Debugger (e.g., gdb, etc.)
◆ Invalid pointer => Kernel Panic
  ✦ Machine halts with (usually) un-helpful console display, much in hex.
  ✦ No nice “segmentation fault” indication.
  ✦ You will use a Virtual Machine (VM) on your laptop for easy recovery from panics

◆ Programming success depends on:
  ✦ Coding correctly and “defensively” the first time,
  ✦ Taking small incremental steps,
  ✦ Using print statements if debugging is necessary.
Some Linux C Examples
(like code you will need to understand)

```c
struct pid *get_task_pid(struct task_struct *task, enum pid_type type)
{
    struct pid *pid;
    rcu_read_lock();
    if (type != PIDTYPE_PID)
        task = task->group_leader;
    pid = get_pid(task->pids[type].pid);
    rcu_read_unlock();
    return pid;
}
EXPORT_SYMBOL_GPL(get_task_pid);
```

```c
pid_t pid_vnr(struct pid *pid)
{
    return pid_nr_ns(pid, task_active_pid_ns(current));
}
EXPORT_SYMBOL_GPL(pid_vnr);
```

```c
pid_t pid_nr_ns(struct pid *pid, struct pid_namespace *ns)
{
    struct upid *upid;
    pid_t nr = 0;
    if (pid && ns->level <= pid->level) {
        upid = &pid->numbers[ns->level];
        if (upid->ns == ns)
            nr = upid->nr;
    }
    return nr;
}
```

Some Linux C Examples
(like code you will need to understand)

```c
char *d_path(const struct path *path, char *buf, int buflen)
{
    char *res;
    struct path root;
    struct path tmp;
    read_lock(&current->fs->lock);
    root = current->fs->root;
    path_get(&root);
    read_unlock(&current->fs->lock);
    spin_lock(&dcache_lock);
    tmp = root;
    res = __d_path(path, &tmp, buf, buflen);
    spin_unlock(&dcache_lock);
    path_put(&root);
    return res;
}

EXPORT_SYMBOL(d_path);
```

Some Linux C Examples
(unlike many functions, that one came with comments)

```c
/**
 * d_path - return the path of a dentry
 * @path: path to report
 * @buf: buffer to return value in
 * @buflen: buffer length
 *
 * Convert a dentry into an ASCII path name. If the entry has been deleted
 * the string " (deleted)" is appended. Note that this is ambiguous.
 *
 * Returns a pointer into the buffer or an error code if the path was
 * too long. Note: Callers should use the returned pointer, not the passed
 * in buffer, to use the name! The implementation often starts at an offset
 * into the buffer, and may leave 0 bytes at the start.
 *
 * "buflen" should be positive.
 */
```
If You Want to Take the Honors Course:

◆ Before Enrolling, get instructor permission:
  ✷ Email to me (smithfd) or Kevin (jeffay) a 1-2 paragraph summary of your background
  • If you have already gained permission, proceed to enroll.
  ✷ If you would like to meet and discuss it, send me email to schedule an appointment.

◆ After Enrolling:
  ✷ Attend one of the workshops with Murray Anderegg to install a Virtual Machine on your laptop
    • Workshops in FB 331 at 3:00 – 4:00 PM on Tuesday, 8/26, and Wednesday 8/27
    • Email to me stating which workshop you plan to attend (or let me know if you cannot attend one of these).
    • Bring laptop to workshops and each class meeting
  ✷ Get a copy of the reference book: