CSCI 6907: Special Topics in Operating Systems

Your Prof: Gabe

Acknowledgements: Some slide material derived from Silberschatz, et al.
Today

- Syllabus and class plan
- Piazza
- Homework – Due Jan 31st
  - This assignment will take 2 weeks, at least
  - start now – tonight!
- Structure – presentations/reading/project
  - will place a sign up sheet online soon – see piazza
- Project
- Project
“High-level”

Cars

Computers
Cars

Computers

```c
/*
 * So far all flags should be taken in the context of the
 * actual invoking thread (they reflect the thread switching
 * _from_ rather than the thread to switch _to_) in which case
 * we would want to use the sched_page flags.
 */
flags = rflags;
switch_thread_update_flags(da, &flags);

if (unlikely(flags)) {
    thd = switch_thread_slopath(curr, flags, curr_spd, rthd_id, da, &ret_code &curr_sched_flags, &tnd_sched_flags);
    /* If we should return immediately back to this
        * thread, and its registers have been changed,
        * return without setting the return value */
    if (ret_code == COS_SCHED_RET_SUCCESS && thd == curr) goto ret;
    if (thd == curr) goto_err(ret_err, "sloopath");
} else {
    next_thd = switch_thread_parse_data_area(da, &ret_code);
    if (unlikely(D == next_thd)) goto_err(ret_err, "data_area\n");
    thd = switch_thread_get_target(next_thd, curr, curr_spd, &ret_code);
    if (unlikely(NULL == thd)) goto_err(ret_err, "get target");
}

/* If a thread is involved in a scheduling decision, we should
 * assume that any preemption chains that existed aren't valid
 * anymore. */
break_preemption_chain(curr);
```
...“low-level”

Wiring diagram for the power supply, starting, charging, engine, front and rear wiper, cigarette lighter, sunroof (G11), sound system and rear demister systems. G10, G11 and G11 Turbo models.
What is an Operating System!?
What is an OS: Analogy

You!

Customer_1

Customer_2

Customer_n
What is an OS: Analogy

You!
Customer\text{\textsubscript{1}}
Customer\text{\textsubscript{2}}
Customer\text{\textsubscript{n}}
What is an OS: Analogy

Hardware

Operating System

Applications

You!

Customer_1

Customer_2

Customer_n
OS as Abstraction: System Layers
Operating System as Abstraction

- "The effective exploitation of his powers of abstraction must be regarded as one of the most vital activities of a competent programmer." - Edsger W. Dijkstra

- Provides abstractions for resources (memory, CPU, disk) and controls application execution
- Provide environment for application execution
- Protection and Isolation
- OS as the provider of system abstractions
Computers as Distributed Systems

“Hardware: The parts of a computer system that can be kicked.”
- Jeff Pesis
OS as Resource Manager

- Control a diverse set of hardware
  - Processors
  - Memory
  - Disks
  - Networking cards
  - Video cards
- Coordinates these hardware resources amongst user programs
  - Want to use resources efficiently!
- OS as a resource manager/multiplexer
AN x64 PROCESSOR IS SCREAMING ALONG AT BILLIONS OF CYCLES PER SECOND TO RUN THE XNU KERNEL, WHICH IS FRANTICALLY WORKING THROUGH ALL THE POSIX-SPECIFIED ABSTRACTION TO CREATE THE DARWIN SYSTEM UNDERLYING OS X, WHICH IN TURN IS STRAINING ITSELF TO RUN FIREFOX AND ITS GECKO RENDERER, WHICH CREATES A FLASH OBJECT WHICH RENDERS DOZENS OF VIDEO FRAMES EVERY SECOND BECAUSE I WANTED TO SEE A CAT JUMP INTO A BOX AND FALL OVER.

I AM A GOD.
What is Operating Systems Research!??
OS Research

- Systematic investigation of novel, interesting, and useful techniques to
  - Better utilize resources
    - System performance, predictability
  - Provide new programming abstractions
    - Enable systems to do things they weren't previously able
      - Recover from faults
      - Tolerate malicious users
    - More easily do the same thing
      - RPC
Why
Operating Systems Research!??
Worse is Better

• Background
  • Lisp: functional programming language
    – Automatic memory management – garbage collection
    – **Simple** semantics
      • lambda, if, numbers, booleans, application (function call), cons
      • ...yet amazingly powerful
  • vs. C
    – Manual memory management (malloc, free)
    – **No** semantics – *anything* can happen...and does!

• Which won? Why?
Worse is Better

- Read Richard Gabriel's piece, then Linus' email
- Write on back:
  - What is Richard's main thesis?
  - Do you agree? Why?
  - What are three examples of worse is better?
  - Any examples of better is better?

- What does this have to do with OS research?
This class...again

- Presenting papers
  - An acquired skill...takes diligent practice
  - If you are worried about this, spend more time on it!
- Reading papers
  - An acquired skill...takes diligent practice
  - Every week...so you will practice this!
- Semester-long project!
The Nucleus

- An example of “better”
- Practice reading a research paper
  - Read up to section 5
- Critically Reading a Research Paper:
  - What is the central point/contributions?
    - 1-3 sentences
  - Three most significant issues/problems/limitations with the approach?
    - Key: critically read the paper
  - Three questions you have about the approach?
    - Often derived from the paper's motivation, design, or evaluation
What/Why
Operating Systems Research!?
How Operating Systems Research!?

Welcome to Special Topics in OS
System Structure

- **System Structure** – How different parts of software
  1) Are separated from each other (*Why?*)
  2) Communicate

- How does a system use
  - dual mode
  - *virtual address spaces*
Monolithic System Structure

- Includes Unix/Windows/OSX
Monolithic System Structure

- Includes Unix/Windows/OSX

When's the last time you tried to get **50 MLOC** to work???

- Hardware (hard drive, NIC, GPU)
Microkernel System Structure

- Moves functionality from the kernel to “user” space
- Communication takes place between user servers using inter-process communication (IPC)

Benefits:
- Easier to add functionality
- More reliable (less code is running in kernel mode)
- More secure

Detriments: performance! (why?)