udp Escuela de Informática y Telecomunicaciones
FACULTAD DE INGENIERÍA

Introducción

Adín Ramírez
adin.ramirez@mail.udp.cl

Sistemas Operativos (CIT2003-1)
1er. Semestre 2015

Adín Ramírez

- Profesor ciencias de la computación
  - Ph.D., Visión por Computador, Kyung Hee University, 2013
  - Áreas: visión por computador, procesamiento de imágenes, reconocimiento de patrones
- Sistemas Operativos (SO)
  - Una nueva experiencia para todos
  - Aprenderemos en conjunto

¿Cuál forma de aprender usas?

Basada en la enseñanza

- El profe debe enseñarme
  - Se vierten contenidos en los alumnos
  - Se pasa la materia
- Se concentra en los **contenidos** (y repetirlos)

Basada en el aprendizaje

- Yo debo aprender
  - Se busca el entendimiento
  - Distintas maneras de aprender para cada uno
- Aprendizaje activo, y enfocado en **entender**

Nuestro enfoque

- Utilizaremos un paradigma basado en el aprendizaje
- Buscaremos un aprendizaje activo
  - Actividades
  - Discusiones
  - Trabajos para cimentar los conceptos
- Desarrollaremos un pensamiento crítico
- Actividades basadas en resultados (siempre con límites de tiempo para los entregables)
  - La teoría es **importante**,
  - pero también poder ponerla en **práctica correctamente**
- Si cambiamos el paradigma, tenemos que cambiar nuestra forma de pensar y de actuar

Esquema general

- SO es un curso denso
- Mucho contenido, nuevos conceptos
  - Abstracción del kernel
  - Interfaz de programación
  - Concurrencia e hilos (hebras)
  - Sincronización de objetos compartidos, y avanzada
  - Scheduling
  - Direcciones de memoria
  - Memoria Virtual
  - Administración de la memoria
  - Almacenamiento

Q: ¿Es eso mucho contenido para un semestre?

A: Sí, sino se administra bien el tiempo

Hábitos de estudio

- Desarrollen una estrategia de estudio tempranamente
- Establezcan buena relación con los compañeros para el desarrollo de la parte práctica (detalles adelante)
- Establezcan un horario para poder leer, programar, y estudiar (detalles sobre el tiempo más adelante)
- **El curso es denso, y hay mucho trabajo** (tanto práctico como teórico), **sino empiezan a tiempo no podrán terminar ni entender todo** lo que tenemos planificado

Créditos

- Según la UDP, 1 crédito = 30 horas
- SO: 6 créditos = 180 horas (semestrales)
  - 1 semestre = ~16 semanas
  - 1 semana = ~11 horas
- 3 horas presenciales, **8 horas de estudio/trabajo independiente** (semanales)
- El trabajo de SO está diseñado considerando ese tiempo para la elaboración de tareas, lecturas, y proyectos.
- Recuerden, los grupos suman horas
  - 2 personas: 16 horas semanales
  - 3 personas: 24 horas semanales
  - ...

  pero hay que considerar la sobrecarga de administración!

Libro de texto

- (Experimental) **Operating Systems: Principles & Practice**, Adnerson & Dahlin
  - Una visión más moderna de sistemas operativos
- (Tradicional) Operating System Concepts, Silberschatz, Galvin, & Gagne
  - Acá también están los temas, cubre más seguridad, visión más antigua
- Tendremos lecturas obligatorias (semanales)
- Muchos conceptos tomados de estos libros
- Consigan un libro **ya!** (muchas opciones)

Objetivos

- Sistemas operativos
  - ¿Qué son?
  - Decisiones de diseño
  - Construcción
- Programación en pareja
  - Diseño, documentación
  - Control de fuente
  - Habilidades blandas
- Presentación de información técnica
  - Escrita
  - Oral (habilidades de presentación)

Clases

- Muchos temas serán cubiertos a través del texto (lean!)
- Faltar a clases afectará tu nota
  - El mapa no es el terreno, las diapositivas no son la clase
  - Si faltas, te pierdes las preguntas y respuestas
  - Tendremos actividades en clase (algunas con nota, asiste para no perderlas)
- Espero que asistan a las clases
  - ¿Recuerdan el paradigma de aprendizaje?
  - Para cimentar el conocimiento y entender, tenemos que conversar y discutir
  - ☹ Pagar para **solo** leer es un mal negocio

Actividades

- Proyectos (tentantivos):
  - Manejo del stack
  - Manejo de hilos
  - Concurrencia
  - Kernel
  - Extensión del kernel
- Ambiente de desarrollo:
  - Máquinas virtuales o en el computador
  - SO: `Minix`
  - Lenguaje: `C`
  - Control de versiones: `git`
  - Documentación en el código: `doxygen`

Programación en equipo

- ¿Por qué?
  - Ayuda a atacar problemas más grandes y complejos
  - Enseña habilidades necesarias que usarán en el trabajo
    - Establecer hitos
    - Establecer un flujo de trabajo productivo
    - Involucrar la *administración* antes que sea demasiado tarde
- Programación en equipo/parejas $\neq$ *ingeniería de software*
  - No hay análisis de requerimientos
  - No hay releases, diseño para crecer, ...
  - No es un ciclo completo de software

Problemas con su compañero

- Alguien tendrá problemas con su compañero
  - Necesitan empezar a involucrarse en la *administración* tempranamente
    - La mayoría de las veces (50 %) se puede solucionar el problema
    - Si el problema no se puede resolver, podemos reducir el daño . . . solo si lo se a tiempo
  - No dejen de lado los problemas, ni a última hora
    - Para ese entonces, ya no no se puede hacer nada

Evaluaciones

- Lecturas semanales, evaluadas en clase:
  - Preguntas y respuestas
  - Actividades en clase
- Reportes
  - ~3, para profundizar el entendimiento de temas
  - Tendremos uno de prueba para que vean como se calificarán (no tendrá punteo)
- Tarea de lectura (detalles durante el semestre)
  - Escojan algún tema interesante
  - Escriban un reporte corto
- Controles
  - ~3, para evaluar el progreso de cada uno

Aclaraciones

- Los informes (reportes, tareas de investigación, y cualquier otro documento) tienen entrega electrónica y física (seguir lineamientos de cada actividad)
- Los informes deben escribirse en LaTeX (y amigos—usando la forma `IEEETrans`)
  - Leer y entender *Guía de documentos técnicos*
  - Entrega impresa del documento, y digital del código fuente (`.tex`) del documento

  Q: No he usado LaTeX, ¿qué hago?

  A: Empieza desde hoy a aprenderlo (hay muchos **recursos en linea**)
- **No se aceptarán entregas tarde** (tolerancia cero)

Ponderación

- La nota es: 30 % + 70 %
- 30 % en el examen final
- 70 % de la nota final está dividida en:
  - Solemnes 30 % (15 % cada uno)
  - Proyectos 40 % (~10 % cada uno)
  - Controles 20 %
  - Tareas, ejercicios y asistencia 10 %

Proyectos

- Cada proyecto está dividido en:
  - 60 % del desarrollo práctico (alcance de los objetivos, desarrollo de software, diseño, completitud, cumplimiento de pruebas, etc.)
  - 40 % informe del proyecto
- Deben de tener azul (sobre 50 % de la nota promedio) de proyectos para poder acceder a examen

Informes

- La ponderación (tentativa) de los informes es:
  - Resumen 5 %
  - Introducción 10 %
  - Cuerpo (contenido) 75 %
  - Conclusiones 10 %
- Se penalizarán faltas de ortografía y de redacción hasta un 20 % de la nota total
- Cada falta de ortografía o de redacción descuenta un punto porcentual
- Editen sus informes antes de ser entregados, ya que el 80 % del trabajo en un documento es la edición, y solo el 20 % es la escritura del documento.

Horarios

- Horario de clases:
  - Cátedra: Martes y Viernes 10:00 a 11:20 (Módulo B)
  - Ayudantía: Miércoles 17:00 a 18:20 (Módulo F)
- Evaluaciones[1]:
  - 1ra Solemne: primera semana de Mayo
  - 2da Solemne: última semana de Junio
  - Examen: Julio
- Consultas:
  - Email: `adin.ramirez@mail.udp.cl`
  - Escuela de Informática y Telecomunicaciones, 2do piso, Edificio Vergara 432 (**Martes y Viernes 14:00 a 17:00, atención a casos especiales con previa cita al email**)

[1] Fechas tentativas

Sitio del curso

- Moodle: `cursos.fic.udp.cl`
- Matricularse en el curso: *Sistemas Operativos*
- Password: *Sistemas operativos 2015* (sensible a mayúsculas y minúsculas)
- El material y las tareas se subirán y entregarán por allí (matricularse oportunamente)