



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2333 — Sistemas Operativos y Redes — 1/2020

Programa de Curso

Actualizado al 17-Abril-2020

Profesor: Cristian Ruz, cruz@ing.puc.cl
Clases: Lunes y Miércoles, módulo 1.
Ayudantías: Viernes, módulo 1.
Requisitos: IIC2343 — Arquitectura de Computadores
Sitio web: <http://iic2333.ing.puc.cl>
Ayudantes: Ricardo Schilling, reschilling@uc.cl, especialista en sistemas operativos
Raúl Álvarez, rsalvarez@uc.cl, especialista en sistemas operativos
Raimundo Martínez, rmartinez7@uc.cl, especialista en redes
Ariel Martínez, amartinez5@uc.cl, especialista en redes

1. Descripción

Cada vez que usamos un dispositivo de cómputo estamos interactuando con un sistema operativo. Es más, hace años que el uso de dispositivos como computadores, móviles, consolas y otros dispositivos de cómputo está regulado no por las capacidades del *hardware*, sino por lo que el sistema operativo nos permite hacer con este *hardware*. Entender el funcionamiento, los componentes y las limitaciones de los sistemas operativos es fundamental al momento de diseñar *software* que pueda aprovechar completamente las capacidades de los instrumentos que utilizamos.

Por otro lado, una buena porción del *software* que programamos está destinado a interactuar con otro *software* que no se encuentran en la misma máquina. Las redes de comunicación de datos permiten que los programas interactúen con otros programas ejecutándose en distintas partes del mundo de manera coordinada a velocidades que parecen instantáneas. Conseguir que una red de alcance mundial funcione coordinadamente requiere el acuerdo de múltiples reglas que no han sido fáciles de definir. Conocer el funcionamiento y configuración de una red de datos permite entender sus limitaciones y es crucial al momento de construir programas que interactúen de manera remota.

2. Objetivo General

A lo largo de este curso el alumno logrará un conocimiento detallado del diseño, composición y el funcionamiento interno de los sistemas operativos y de las redes de comunicación de datos. Este conocimiento permitirá a los alumnos ser capaces de diseñar *software* que aproveche las características tanto del sistema operativo como de una infraestructura de red, comprender las razones que limitan su funcionamiento, y cómo aprovecharlas o mejorarlas.

A través de este curso conocerán características que comparten sistemas operativos de uso común como Windows y Linux, y tendrán mejores herramientas para enfrentarlos y entender sus ventajas y desventajas. En el aspecto de redes estudiaremos el funcionamiento de redes de pequeño y gran tamaño como la Internet de manera que este conocimiento les permita construir mejores aplicaciones distribuidas.

3. Competencias

Al finalizar el curso el alumno será capaz de:

1. **Comprender** los subsistemas que conforman un sistema operativo, sus objetivos, decisiones de diseño y mecanismos de implementación.

2. **Analizar** el funcionamiento de algoritmos de *scheduling*, memoria virtual y almacenamiento en un sistema operativo.
3. **Construir** subsistemas de bajo nivel que puedan interactuar con componentes de *software* destinados al usuario a través de una interfaz precisa.
4. **Comprender** los componentes de *hardware* y *software* que se utilizan en la construcción de redes de datos, y sus mecanismos de operación.
5. **Evaluar** el funcionamiento de redes de datos mediante el análisis de capas y componentes.
6. **Diseñar** redes de datos en base a un conjunto de restricciones y requisitos.
7. **Construir** *software* que utilice redes de datos para implementar una aplicación distribuida.

4. Contenido

- | | |
|--|---|
| 1. Funcionamiento de Sistemas Operativos | 5.3) Redes e Internet |
| 1.1) Estructura de un sistema operativo | 5.4) Modelos de referencia: OSI, TCP/IP |
| 1.2) Evolución de sistemas operativos | 6. Capa de Aplicación |
| 1.3) Llamadas al sistema | 6.1) Comunicación de Procesos y Sockets |
| 2. Administración de procesos | 6.2) HTTP y la Web |
| 2.1) Procesos: operaciones y comunicación | 6.3) Protocolos: FTP, SMTP, DNS |
| 2.2) Threads: librerías y multithreading | 7. Capa de Transporte |
| 2.3) Sincronización: sección crítica y primitivas | 7.1) Multiplexión y Demultiplexión |
| 2.4) Planificación: algoritmos de <i>scheduling</i> | 7.2) UDP |
| 3. Administración de memoria | 7.3) Transmisión Confiable y TCP |
| 3.1) Asignación, direccionamiento y <i>swapping</i> | 8. Capa de Red |
| 3.2) Paginación y segmentación | 8.1) Modos de conexión |
| 3.3) Memoria virtual: paginación y reemplazo | 8.2) Direccionamiento IP, IPv6, NAT, ICMP |
| 4. Administración de sistemas de almacenamiento | 8.3) DHCP |
| 4.1) Disco: Estructura, acceso y planificación | 8.4) Algoritmos de enrutamiento, e Internet |
| 4.2) Sistemas de archivos: estructuras y administración de espacio | 9. Capa de Enlace |
| 5. Modelos de redes | 9.1) Detección y corrección de errores |
| 5.1) Evolución de las redes de computadores | 9.2) Protocolos de Acceso al Medio (MAC) |
| 5.2) Tipos de redes y modelos de comunicación | 9.3) Switches y LAN: ARP, Ethernet, VLANs |

5. Metodología

Este curso utilizará una metodología *blended* en que se combinarán clases expositivas con actividades prácticas.

El material de clases estará disponible de manera *online* para que pueda ser leído y consultado antes y durante la clase expositiva. El material será complementado con videos acerca de temas específicos. En algunas clases, que serán previamente informadas, se realizarán actividades prácticas.

El horario de los días viernes se utilizará para actividades de ayudantía y clases recuperativas, que serán debidamente notificadas.

En caso de restricciones a las clases presenciales, se organizarán sesiones de clases remotas. El detalle será informado a través del canal oficial en el sitio de Canvas.

6. Evaluación

Se efectuarán evaluaciones escritas (3), evaluaciones *online* (14), y evaluaciones prácticas (6).

Sobre las evaluaciones escritas

Las evaluaciones escritas se realizarán y se entregarán a través de Canvas

Las fechas de las evaluaciones escritas son las siguientes:

Evaluación	Fecha	Duración	Temática
I_1	W 22 de Abril 2020, 8:00-20:00	75 min	Comprensión y análisis de aspectos de sistemas operativos, administración de procesos y administración de memoria.
I_2	W 20 de Mayo 2020, 8:00-20:00	75 min	Comprensión y análisis de aspectos de sistemas de archivos, funcionamiento de redes, y capa de aplicación.
E	M 30 de Junio 2020, 8:00-20:00	75 min	Comprensión y análisis de aspectos de capa de transporte, red, y enlace.

No se borra ninguna interrogación. En caso de inasistencia a I_1 o a I_2 , se dará la opción de recuperar esta evaluación inmediatamente a continuación de E .

En caso de no poder rendir alguna interrogación, I_1 , I_2 ó E , se dará la opción de rendirla en una fecha posterior al fecha de finalización de clases.

Los días de interrogaciones **no habrá clases**. En el horario 8:30-9:50 se habilitará un enlace Zoom para realizar consultas sobre enunciado. Las respuestas se entregarán en ese momento y se publicarán a través de una *issue* de un foro Github. Fuera de este horario se recibirán consultas via *issues* de Github, y se actualizarán hasta las 20:00.

Sobre las evaluaciones online

Las evaluaciones *online* se publicarán y serán efectuadas y evaluadas dentro de la plataforma Canvas,

Actividad	Temática
$AC_{SO} = AC_1, \dots AC_7$	Comprensión y análisis de aspectos de sistemas operativos, administración de procesos, administración de memoria, y administración de archivos.
$AC_R = AC_8, \dots AC_{14}$	Comprensión y análisis de aspectos de funcionamiento de redes, capa de aplicación, transporte, red y enlace

Se eliminarán las dos **tres** peores notas de actividades entre $AC_1, \dots AC_7$, y las dos **tres** peores notas entre $AC_8, \dots AC_{14}$

Sobre las evaluaciones prácticas

Se realizarán 6 evaluaciones prácticas. Ellas tienen como objetivo evidenciar las competencias 3, 5, 6 y 7 del curso.

Tarea	Publicación	Entrega	Temática
T_1	V 27-Marzo	W 8-Abril	Construcción de <i>software</i> de manejo de procesos en C
T_2	L 13-Abril	L 27-Abril	Construcción de <i>software</i> de manejo de procesos en C
P_1	V 24-Abril	M 13-Mayo	Construcción de <i>software</i> de bajo nivel en C
P_2	V 15-Mayo	M 3-Junio	Construcción de <i>software</i> con transmisión de datos de aplicación en C
T_3	L 8-Junio	Lunes 15-Junio	Evaluación de funcionamiento de redes mediante monitoreo de tráfico
T_4	V 19-Junio	V 25-Junio	Diseño de red mediante <i>software</i> de simulación

El día 8 de Junio se realizará una actividad práctica en la clase. Es la única fecha cuya **asistencia con computador es obligatoria**.

Sobre el cálculo de la nota final

Se considerará una nota de sistemas operativos N_{SO} , y una nota de redes N_R . Para cada una de ella se pondera un 60 % de las evaluaciones prácticas (competencias 3, 5, 6 y 7) y 40 % de las evaluaciones teóricas (competencias 1, 2 y 4)

- $N_{SO} = 0,3 \times P_1 + 0,15 \times T_1 + 0,15 \times T_2 + 0,16 \times I_1 + 0,08 \times I_2 + 0,16 \times AC_{SO}$
- $N_R = 0,3 \times P_2 + 0,15 \times T_3 + 0,15 \times T_4 + 0,16 \times E + 0,08 \times I_2 + 0,16 \times AC_R$

En caso de obtener un promedio $\geq 3,0$ entre T_1 y T_2 , se considerará tanto a T_1 como a T_2 igual a $\max(T_1, T_2)$

Los requisitos de aprobación se basan en lograr una nota mínima de 3,95 en el conjunto de instrumentos de evaluación para cada grupo de competencias, y una ponderación mínima de 3,95 en el conjunto del curso. Se deben cumplir 2 de las siguientes 4 condiciones:

- Competencias 1 y 2: $\frac{0,16 \times AC_{SO} + 0,16 \times I_1 + 0,08 \times I_2}{0,4} \geq 3,95$
- Competencias 4 y 6: $\frac{0,16 \times AC_R + 0,16 \times E + 0,08 \times I_2}{0,4} \geq 3,95$
- Competencia 3: $\frac{0,3 \times P_1 + 0,15 \times T_1 + 0,15 \times T_2}{0,6} \geq 3,95$
- Competencias 5, 6 y 7: $\frac{0,3 \times P_2 + 0,15 \times T_3 + 0,15 \times T_4}{0,6} \geq 3,95$

La nota final, N_F , se calcula como:

$$N_F = \begin{cases} 0,5 \times N_{SO} + 0,5 \times N_R & \text{si se cumplen todos los requisitos} \\ \min\{0,5 \times N_{SO} + 0,5 \times N_R, (3,9)\} & \text{en otro caso} \end{cases}$$

7. Bibliografía

El curso está preparado de manera que el material de clases y las referencias que se hagan durante ellas sean suficiente para comprender los contenidos. Sin embargo, aquellos que desean profundizar más en los contenidos están invitados a hacerlo. Las principales fuentes que se han utilizado para preparar el material son:

1. Andrew S. Tanenbaum, Herbert Bos. *Modern Operating Systems*, 4th Edition. Pearson. Dec 2012. ISBN 978-0-133-59162-0.
2. James F. Kurose, Keith W. Ross. *Computer Networking. A Top-Down Approach*. 6th Edition. 2013. Pearson. ISBN-13: 978-0-13-285620-1.

Las siguientes son fuentes complementarias, algunas de ellas *online*:

1. Abraham Silberschatz, Peter Galvin, Greg Gagne. *Operating Systems Concepts*, 9th Edition. John Wiley & Sons, Inc. Dec 2012. ISBN 978-1-118-06333-0. <http://os-book.com/>
2. Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau. *Operating Systems: Three easy pieces* Arpaci-Dusseau Books, August 2018. Versión 1.00. <http://pages.cs.wisc.edu/~remzi/OSTEP/>
3. Olivier Bonaventura. *Computer Networking. Principles, Protocols and Practice*, Online (*open source*, in edition) <https://www.computer-networking.info/2nd/html/>

8. Compromiso del Código de Honor

Este curso adscribe el Código de Honor establecido por la Universidad, el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso de que exista colaboración permitida con otros alumnos, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como alumno es su deber conocer el Código de Honor (www.uc.cl/codigodehonor).

Política de Integridad Académica del Departamento de Ciencia de la Computación

Se espera que los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile mantengan altos estándares de honestidad académica, acorde al Código de Honor de la Universidad. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Pregrado de la Escuela de Ingeniería (Disponible en SIDING, en la sección Pregrado/Asuntos Estudiantiles/Reglamentos/Reglamentos en Ingeniería/Integridad Académica).

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho **individualmente** por el alumno, **sin apoyo en material de terceros**. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un alumno copia un trabajo, o si a un alumno se le prueba que compró o intentó comprar un trabajo, **obtendrá nota final 1.1 en el curso** y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral. Por “copia” se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso que corresponda a “copia” a otros alumnos, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, **siempre y cuando se incluya la referencia correspondiente**.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile (<http://admisionyregistros.uc.cl/alumnos/informacion-academica/reglamentos-estudiantiles>). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.