



Universidad
de O'Higgins

VICERRECTORÍA ACADÉMICA

UNIDAD DE INNOVACIÓN Y GESTIÓN CURRICULAR

PROGRAMA DE ACTIVIDAD CURRICULAR

1) IDENTIFICACIÓN DE LA ACTIVIDAD CURRICULAR			
NOMBRE DE LA ACTIVIDAD CURRICULAR	Programación de Software de Sistemas		
UNIDAD ACADÉMICA	Escuela de Ingeniería		
CARRERA	Ingeniería Civil en Computación	TIPO DE ACTIVIDAD	Obligatoria
CÓDIGO	COM2002	SEMESTRE	4to Semestre
CRÉDITOS SCT-Chile	6	SEMANAS	18
TIEMPO DE DEDICACIÓN SEMANAL			
TIEMPO DE DEDICACIÓN TOTAL	TIEMPO DE DOCENCIA DIRECTA	TIEMPO DE TRABAJO AUTÓNOMO	
9 horas	4,5 horas	4,5 horas	
REQUISITOS			
PRE-REQUISITOS	CO-REQUISITOS		
Algoritmos y Estructuras de Datos	No tiene		

2) DESCRIPCIÓN DE LA ACTIVIDAD CURRICULAR
<p>El curso tiene como propósito brindar a las/los estudiantes una comprensión sólida y práctica de la programación a bajo nivel, utilizando el lenguaje de programación C en el entorno del Sistema Operativo Linux. A través de este curso, las/los estudiantes desarrollarán habilidades fundamentales en la gestión eficiente de memoria, manejo de procesos, programación concurrente, y uso de librerías estándar.</p>



Universidad
de O'Higgins

VICERRECTORÍA ACADÉMICA

UNIDAD DE INNOVACIÓN Y GESTIÓN CURRICULAR

3) RESULTADOS DE APRENDIZAJE

1. Escriba y entienda programas eficientes en lenguaje de programación C, utilizando las funciones básicas que provee el Sistema Operativo Linux tanto para la administración de sistemas, para hacer programas eficientes y para programación concurrente.
2. Maneje los conceptos básicos de la programación de sistemas: arquitectura de computadores, direcciones de memoria, notación hexadecimal, representación de enteros, bits.
3. Desarrolle software de sistemas para Linux usando las funciones de manejo de memoria, Entrada/Salida, sistema de archivos, sockets y threads.

4) UNIDADES DE APRENDIZAJE Y CONTENIDOS

UNIDAD 1: Programación eficiente en C

- Representación de los datos. (enteros, bits, hexadecimal)
- Manejo de memoria (global, local dinámica).
- Strings.
- Arreglos, punteros.
- Estructuras (struct).
- Profiling y debugging.
- Setjmp/longjmp.
- Macros, switch, tabla de saltos

UNIDAD 2: Programación concurrente

- Threads y C (POSIX threads).
- Sincronización: exclusion mutua, semáforos, monitores.
- Ejemplos clásicos: productor/consumidor, filósofos.

UNIDAD 3: Programación de procesos

- Uso de Entrada y salida, programación de procesos y sockets
- Caso de estudio: Sistema operativo Linux

UNIDAD 4: Librerías estándares

- Herramientas para la programación de sistemas (Shell Programming, Perl Programming, Web Programming (cgi-bin, php))

RECURSOS Y METODOLOGÍAS DE ENSEÑANZA APRENDIZAJE

Metodologías de aprendizaje

- Clases expositivas
- Experiencia de aprendizaje basado en problemas y tareas aplicadas
- Elaboración de informes en tareas individuales
- Análisis y discusión de casos teóricos y prácticos.



Universidad
de O'Higgins

VICERRECTORÍA ACADÉMICA

UNIDAD DE INNOVACIÓN Y GESTIÓN CURRICULAR

5) CONDICIONES DE EVALUACIÓN Y APROBACIÓN

1. LA ASIGNATURA SE APRUEBA SI: $NF \geq 4.0$ siempre y cuando $NC \geq 4.0$ y $NAC \geq 4.0$
2. LA ASIGNATURA SE EXIME DEL EXAMEN SI: $NP \geq 5.5$

La nota final del curso (NF) está compuesta por una Nota de Cátedra (NC) y una Nota de Actividades Complementarias (NAC) con las siguientes ponderaciones:

$$NF = 0.5 NC + 0.5 NAC$$

La Nota de Cátedra (NC) está compuesta por las Nota de Presentación (NP) y Examen (NE) con las siguientes ponderaciones:

$$NC = 0.6*NP + 0.4*NE.$$

La Nota de Presentación (NP) está compuesta por las notas de los tres Controles de Cátedra (NCC), con las siguientes ponderaciones:

$$NP = 1/3*NCC1 + 1/3*NCC2 + 1/3*NCC3$$

La NAC está compuesta por el promedio de las Notas de Tareas (NT) con las siguientes ponderaciones:

$$NAC = 1/3*NT1 + 1/3*NT2 + 1/3*NT3$$

EXENCIÓN AL EXAMEN

1. Cualquier estudiante cuya NP sea superior o igual a 5.5, está exento de rendir el Examen.
2. Los/las estudiantes exentos/as del examen recibirán como nota de examen su nota de presentación NP.
3. Si los/las estudiantes exentos desean, pueden de igual forma rendir el examen, en cuyo caso su EX se considerará solo si mejora su calificación final del curso.
4. El examen **no** reemplaza la menor nota que conforma la NP.
5. Si posterior al Examen el/la estudiante no ha cumplido con los criterios de aprobación del curso, tendrán derecho a rendir el examen recuperativo solo los/as estudiantes que tengan una NC igual a 3.7 a 3.9. La aprobación de este ex. Rec dará como resultado que la $NC=4.0$. (**No nota final**)

INASISTENCIA A EVALUACIONES

Toda inasistencia a un Control de Cátedra y no entrega de una Tarea será calificada con la nota mínima (1,0). Sólo podrá ser reemplazada en caso de haber justificado su ausencia frente a las unidades respectivas de la Universidad.



Universidad
de O'Higgins

VICERRECTORÍA ACADÉMICA

UNIDAD DE INNOVACIÓN Y GESTIÓN CURRICULAR

FECHAS DE EVALUACIÓN

- Control de Cátedra 1:
 - o Semana 5, Martes 10 de Septiembre (Horario Cátedra)
- Control de Cátedra 2:
 - o Semana 10, Miércoles 16 de Octubre (Horario Ayudantía)
- Control de Cátedra 3:
 - o Semana 14, Miércoles 13 de Noviembre (Horario Ayudantía)
- Tarea 1:
 - o Publicación: Semana 4, Miércoles 4 de Septiembre
- Tarea 2:
 - o Publicación: Semana 8, Miércoles 2 de Octubre
- Tarea 3:
 - o Publicación: Semana 12, Miércoles 30 de Octubre

Observación: Semana 6 (16 al 20 de Septiembre) corresponde a receso académico

6) BIBLIOGRAFÍA OBLIGATORIA	
REFERENCIAS BIBLIOGRÁFICAS	TIPO DE RECURSO
Brian Kernighan, Dennis Ritchie. The C Programming Language. Prentice Hall. 2nd Edition, 1988.	Digital

7) BIBLIOGRAFÍA COMPLEMENTARIA	
REFERENCIAS BIBLIOGRÁFICAS	TIPO DE RECURSO
N/A	N/A