**LABORATORIO III**

Ingeniería en Informática

Modalidad: Presencial

Carga horaria total: 68 horas

**Profesor: Fabio DellaRosa**

**OBJETIVOS**

* Adquirir los conocimientos necesarios para realizar la programación de la interfaz entre el kernel de UNIX y software de aplicación que se ejecuta en su entorno. Esta programación se desarrollará en lenguaje C.
* Estudiar las principales llamadas al sistema (System calls), que tienen que ver con el manejo de archivos, procesos y los distintos mecanismos de sincronización y comunicación entre procesos
* Analizar algunas librerías de las subrutinas más importantes provistas con el sistema UNIX.
* Contribuir al desarrollo de las siguientes competencias:

| Competencia | Relación con la asignatura |
| --- | --- |
| Identificación, formulación y resolución de problemas de ingeniería en informática | En esta asignatura el estudiante afronta problemas de ingeniería en informática, a través de los requerimientos que le genera la concepción, diagramación y programación de algoritmos de procesamientos de datos en tiempo real,manejo de archivos, semáforos, cola de mensajes, Thread, desarrolla la capacidades de identificación, formulación y resolución problemas que encuentra a lo largo de la disciplina. |
| Utilización de técnicas y herramientas de aplicación en la ingeniería en informática | Debe utilizar las técnicas y herramientas de esta asignatura de programación, aplicando los mejores criterios para la eficiencia de procesamiento y de recursos a alcanzar con los algoritmos, como eficiencia en los tiempos y calidad de desarrollo en los que debe articular con compañeros de clase, y en el futuro con compañeros de profesión. |
| Desempeño en equipos de trabajo | Mediante distintos trabajos grupales se fomenta el trabajo colaborativo, para generar el intercambio de ideas, encontrar diferentes puntos de vista sobre posibles resoluciones de las situaciones problemáticas, y lograr profundizar en el desarrollo de habilidades de argumentación de ideas, permitiendo la discusión y el intercambio, la negociación, y la importancia de llegar a acuerdos. |

**CONTENIDOS MÍNIMOS**

Unix: Conceptos básicos. Recursividad y eventos. Excepciones y concurrencia. Representación de datos en memoria. Estrategias de implementación. Manejo de memoria en ejecución. Uso de heurísticas en memoria. Resolución de Problemas y Algoritmos. Sistema de archivos. Protección. Procesos. Señales. Comunicación entre procesos. Cola de mensajes. Semáforos. Memoria compartida. IPC. Programación en paralelo: Thread. Sockets.

**PROGRAMA ANALÍTICO**

**Unidad 1: Linux y el lenguaje C**

**Módulo 1: Introducción**

Comandos Básicos. Sentencia de compilación (Uso del GCC). Manejo de memoria, representación de datos y eventos. Estructuras de datos.

**Unidad 2: Procesos**

**Módulo 2: Procesos**

Estructura jerárquica de procesos. Llamadas al sistema para la ejecución de un programa. Conceptos de parámetros y argumentos en procesos. Rendón, generación de datos. Llamadas al sistema para finalizar la ejecución de procesos.

**Unidad 3: Archivos**

**Módulo 3: Archivos**

Apertura y creación de Archivos. Manejo de atributos de archivos a través de llamadas al sistema. Manejo de errores generados por las llamadas al sistema. Cadenas de texto en lenguaje C. Comando sprintf.

**Unidad 4: Comunicación entre procesos - IPC**

**Módulo 4: POSIX**

Introducción al estándar POSIX. Real-Time, procesamientos de datos en tiempo real. Concepto de la librería IPC en el lenguaje C. La región crítica, propiedades de acceso exclusivo.

**Módulo 5: Semáforos**

Semáforos. Funciones bloqueantes. Esquema Productor-Consumidor. Sincronizar procesos a través de semáforos. Resolución de casos reales.

**Módulo 6: Memoria Compartida**

Comunicación entre procesos utilizando la memoria compartida. Mecanismos de acceso exclusivo a la memoria compartida. Condición de carrera sobre las variables. Concurrencia.

**Módulo 7: Cola de Mensajes**

Cola de mensajes. Protocolo de creación de mensajes y eventos.

Utilización de librerías en lenguaje C. Introducción a la compilación con el comando MAKE.

**Unidad 5: Thread**

**Módulo 8: Programación en paralelo.**

Introducción a Thread. Compilación con la librería pthread. Mutex. Puntero a función.

**Módulo 9: Multi-Threading.**

Integración de Thread y cola de mensajes. Multi - Threading.

**Unidad 6: Señales**

**Módulo 10: Uso de Señales**

Tipos de señales. Función signal. Función kill.

**Unidad 7: Socket**

**Módulo 11: Comunicación en red**

Canales de comunicación. Protocolos. Introducción a Socket en C. Arquitectura Cliente / Servidor.

**BIBLIOGRAFÍA OBLIGATORIA**

* Linux Professional Institute. Recuperado de: <https://www.lpi.org/>
* Pacheco, P. (2011). *An introduction to parallel programming*. Amsterdam: Morgan Kaufmann.
* Vallejo Fernández, D., González Morcillo, C. y Albusac Jiménez, J. A. (2016). Programación Concurrente y Tiempo Real. España: CreateSpace Independent Publishing Platform
* Wall, K. (2000). Programación en LINUX. Con ejemplos. Buenos Aires: Prentice Hall.
* Obtención de números aleatorios en C: Recuperado de:

<http://www.chuidiang.org/clinux/funciones/rand.php>

* Procesos e Hilos en C de Unix/Linux. Recuperado de:

<http://www.chuidiang.org/clinux/procesos/procesoshilos.php>

* Señales y Alarmas en C de Unix/Linux. Recuperado de:

<http://www.chuidiang.org/clinux/senhales/senhales.php>

**BIBLIOGRAFÍA COMPLEMENTARIA**

* Ejemplos de JAVA y C/ Linux, disponible en <http://www.chuidiang.org/>

**ACTIVIDADES DE FORMACIÓN PRÁCTICA**

Las actividades prácticas planteadas en la asignatura permitirán el desarrollo de habilidades y competencias. Para ello:

* Se le solicitará al estudiante una activa participación, ya que deberá realizar actividades de manera individual o grupal.
* La formación práctica consiste en desarrollar y resolver los ejercicios y situaciones planteadas por el profesor, con el fin de entender y aplicar los conceptos desarrollados en la teoría.
* Se incluirá, además, la resolución de casos de situaciones reales y problemas vinculados a la carrera, emulando las situaciones a las que se enfrenta un profesional.

La asignatura prevé actividades prácticas supervisadas por el profesor.

* Además de los ejercicios prácticos propuestos en cada módulo, los estudiantes deberán resolver en forma obligatoria aquellos designados por el profesor, quien luego los corrige para otorgarle una retroalimentación de su desempeño al estudiante.
* Práctica en máquina: comandos del sistema operativo, obtención de números aleatorios, compilación, gcc de linux.
* Práctica sobre administración de recursos del sistema operativo: Memoria compartida.
* Ejercicio práctico integrador: cola de mensajes, thread/mutex, procesos. Situaciones reales.
* Actividad experimental en máquina: Manejo de señales. Ejercitación con timer, memoria compartida.
* Ejercicios que vinculan (en forma acotada) situaciones reales, como ser:
  + sistemas de peaje y cola de vehículos
  + sistemas de cajero y banco
  + sistemas de juego: bingo, juegos de dados, etc.

**Desarrollo de las competencias**

Las competencias se desarrollan a través de las actividades prácticas, en forma acumulativa junto con las otras asignaturas de la carrera. A continuación, se incluye una tabla con las actividades que aportan al desarrollo de cada competencia relacionada con esta asignatura:.

| Competencia | Actividades |
| --- | --- |
| Identificación, formulación y resolución de problemas de ingeniería en informática | TP I : Formular y resolver una Cola de mensajes, thread/mutex, procesos. Situaciones reales  Act. Aplicar manejo de señales en ejercicios con timer y memoria compartida.  Act. Resolver ejercicios que vinculan situaciones reales como: sistemas de peaje y cola de vehículos, sistemas de cajero y banco,sistemas de juego: bingo, juegos de dados, etc. |
| Utilización de técnicas y herramientas de aplicación en la ingeniería en informática | Act.: Resolución de ejercicios en lenguaje C con sistemas operativos Linux |
| Desempeño en equipos de trabajo | Act. grupal: TP “Procesos”  Act. grupal de debate: Programación en paralelo  TP I: Cola de mensajes, thread/mutex, procesos. Situaciones reales |

**DISTRIBUCIÓN DE LAS ACTIVIDADES FORMATIVAS**

Se prevé la siguiente distribución de actividades y cargas de estudio:

* La carga horaria de esta asignatura es de 68 horas distribuidas en las semanas que comprende el semestre académico.
* En el Calendario correspondiente a cada ciclo lectivo, las fechas de su inicio y final (incluyendo su examen final) podrá consultarse en la web de la Facultad de Ingeniería.

**METODOLOGÍA DE EVALUACIÓN Y EVIDENCIAS DEL APRENDIZAJE**

Las habilidades y competencias conforman parte esencial de la formación del profesional en Informática. Estas se demuestran a partir de las actividades, por lo tanto el acento de las evaluaciones estará sobre las actividades más que en la expresión de conocimientos.

Las actividades que involucran: creación, desarrollo, análisis, y resolución, son las que más evidencian el estado del proceso de aprendizaje del estudiante.

Como la resolución de casos reales es un trabajo iterativo, que requiere proponer y ensayar diferentes alternativas, y luego su evaluación, corrección y/o replanteo de nuevas alternativas. Por lo que se valorarán no sólo los resultados, sino también la actitud de exploración de alternativas de resolución y desarrollo, la originalidad y el aspecto personal del trabajo de cada estudiante.

Las evaluaciones se materializan a través de:

* exámenes escritos
* exámenes orales
* actividades de resolución de problemas

En el caso de las competencias para las cuales esta asignatura contribuye a desarrollar se detallan a continuación las evidencias particulares para cada una:

| Competencia | Evidencias |
| --- | --- |
| Identificación, formulación y resolución de problemas de ingeniería en informática | Para lograr código de calidad y eficiente, el estudiante debe aplicar los conceptos de Integridad, Claridad, Sencillez, Eficiencia, Modularidad y Generalidad y que se verifiquen en su desarrollo. Con esta base, el estudiante identifica los requerimientos de un programa computacional a desarrollar en lenguaje ANSI C y selecciona en forma adecuada las herramientas a utilizar, junto con su modo de uso. El estudiante propone diversas soluciones a un problema planteado pero respetando las técnicas y métricas de programación en C. |
| Utilización de técnicas y herramientas de aplicación en la ingeniería en informática | Con las herramientas de la asignatura, mediante las indicaciones y métricas a seguir, en conjunto con el armado del entorno de trabajo y principalmente de compilación (estándar c89 y c90), para la creación y manipulación de programas en ANSI C el estudiante adquiere los conceptos fundamentales en dicha asignatura. Se busca lograr por medio de las técnicas enseñadas resolución de problemas y programas de alta calidad sin lugar a errores, con foco en su modularidad y manutención. |
| Desempeño en equipos de trabajo | El estudiante participa activamente en el desarrollo de las actividades grupales y en las exposiciones grupales. |

Estas mismas competencias completan su desarrollo a lo largo de toda la carrera a través de aquellas asignaturas con las que se relacionan específicamente según se indica en sus syllabus respectivos.

**REQUISITOS DE APROBACIÓN**

La aprobación de la asignatura requerirá aprobar su Cursado y su Examen Final, en los que deberá demostrar conocimiento de los contenidos y el desarrollo de las competencias asociadas a esta asignatura. El grado esperado de desarrollo de las competencias estará acorde a la posición relativa de esta asignatura en el plan de estudios.

Es obligatorio tener una asistencia al 75% de las clases.

**Requisitos para la aprobación del cursado**

La evaluación de cursado por parte del profesor se realiza mediante una rúbrica con los siguientes coeficientes de ponderación:

| Actividad | Ponderación en % |
| --- | --- |
| Completar la totalidad de las actividades prácticas de cada módulo individuales y/o grupales, realizando las diferentes tareas que se proponen en cada uno de los mismos | 30 |
| Concepto del profesor referente a la participación del estudiante en el curso y al logro de la adquisición de las habilidades necesarias para evidenciar las competencias vinculadas a la asignatura | 15 |
| Aprobar dos exámenes parciales, | 55 |
| Total | 100% |

* Los exámenes parcial se aprueban con una nota igual o superior a 4 (cuatro). Un porcentaje menor de respuestas correctas implicará la reprobación del examen.
* Las evaluaciones durante el cursado serán de carácter formativo e incluirán una retroalimentación del profesor.
* Cualquier obligación académica no cumplida o reprobada podrá ser recuperada. Los exámenes de recuperación de parciales serán tomados dentro del período de cursada o dentro del período que el calendario académico indique para la recuperación.

La aprobación del cursado habilitará al estudiante a rendir el examen final.

**Requisitos para la aprobación del examen final**

La evaluación de la asignatura se completa mediante la evaluación de un Examen Final.

El profesor formulará sus exámenes de forma tal que evalúen que el estudiante haya alcanzado los objetivos de aprendizaje de la asignatura. Se incluirá en la evaluación los temas indicados en el Syllabus vigente de la asignatura, tanto en su faz teórica como en su faz práctica.

Las evaluaciones comprenderán preferentemente diversas modalidades y tipologías de preguntas y actividades que permitan evaluar los conocimientos, habilidades y competencias del estudiante.

Las evaluaciones y la calificación son siempre individuales.

Los exámenes se aprueban con una nota mínima de 4 (cuatro), que implica mayor porcentaje de respuestas correctas.