



UNIVERSIDAD NACIONAL DE LA MATANZA
Departamento de Ingeniería
e Investigaciones Tecnológicas

Cátedras de:

Sistemas de Computación II (Plan 1997)

Sistemas Operativos (Plan 2009)

Jefe de Cátedra: Fabio E. Rivalta

Equipo de Docentes: Boettner F., Catalano L.,
de Lizarralde R, Villamayor A

Auxiliares docentes: Loiacono F., Gariup G, Hirschfeldt D.,
Rodriguez A., Piubel F., Barillaro, S., Barbieri D.

PLANIFICACIÓN Y

GUÍA DE TRABAJOS PRÁCTICOS

(Segundo cuatrimestre 2016)

Contenidos:

- Reglamento, Programa y Planificación.
- Comentarios Previos
- Introducción al Diseño de Software.
- Guía de Trabajos Prácticos Grupales.

DEPARTAMENTO: *Ingeniería e Investigaciones Tecnológicas***ASIGNATURAS:****Sistemas de Computación II
Sistemas Operativos****1. OBJETIVOS.****1.1. OBJETIVOS DEL CURSO.**

- Brindar los conceptos fundamentales y su respectiva actualización tecnológica sobre los Sistemas Operativos.
- Facilitar una actualización sobre las terminologías, y desarrollos tecnológicos sobre Sistemas Operativos Modernos.

1.2. OBJETIVOS DE APRENDIZAJE.

- Que el alumno adquiera el dominio de conceptos básicos y actualizados sobre los Sistemas Operativos e introducir los lineamientos generales de nuevos desarrollos tecnológicos en estos temas.
- Generar una concepción global y un enfoque selectivo para las soluciones algorítmicas de los diferentes problemas que ocurren dentro de un computador y la correcta utilización del mismo.

1.3. META OPERATIVA:

- Se tratara que el alumno al finalizar la materia logre:
- Adquirir el vocabulario y usarlo con precisión.
- Conocer en forma amplia y general la misión y funcionamiento de los componentes de los Sistemas Operativos de un computador.
- Analizar y evaluar por si mismo un Sistema Operativo de cualquier equipo existente en plaza.
- Desarrollar en el Alumno, el interés por la investigación, usando libros y publicaciones propuestas por el Docente.
- Crear en el Alumno, una capacidad de resolución de problemas mediante una adecuada ejercitación práctica.
- Motivar en los alumnos a proponer algunos temas de interés para desarrollar o investigar o encontrar diferentes soluciones a los mismos.

2. ALCANCES.

Los temas a tratar contemplarán básicamente los módulos temáticos que propondrá la cátedra. Su profundidad abarca la extensión de todos los temas específicamente mencionados en el Programa de SISTEMAS DE COMPUTACIÓN II (Plan 1997) y SISTEMAS OPERATIVOS (Plan 2009) vigentes.

PROGRAMA ANALÍTICO.

CONTENIDOS TEÓRICOS Y PRÁCTICOS DE LAS ASIGNATURAS: SISTEMAS DE COMPUTACIÓN II (Plan 1997) y SISTEMAS OPERATIVOS (Plan 2009) – VERSIÓN 2016

Módulo 1: Generalidades de los sistemas operativos

Conceptos de Arquitecturas CISC y RISC. Conceptos de Microprogramación. Conceptos del Lenguaje Assembler.

Interacción con el Sistema Operativo. Limitaciones del Hardware de las Computadoras.

Introducción a los SO. Clasificación. Conceptos fundamentales y conceptos básicos de SO.

Terminología, definiciones y funciones de SO.

Características comunes a todos los SO. Organización y estructura interna de los sistemas operativos.

Componentes mínimos de un SO.: El shell, los administradores del SO., el Kernel o núcleo.

Prestaciones y servicios de los SO.

Módulo 2: Procesos

Definición y concepto de proceso.

Estados de un proceso. Diagrama de estados. Ciclo de vida de un proceso. Transiciones de estado.

Las operaciones sobre un proceso: creación, manipulación y muerte de un proceso.

El control de un proceso. Estructuras de control del sistema operativo.

Tipos de procesos: los procesos pesados y livianos, hilos o hebras (Threads).

Implementación de hilos (Threads). La creación y ejecución de los Threads. Estado de los threads. Uso de los hilos. Sistemas operativos "multithreaded": aspectos del diseño e implementación de paquetes de Threads. El Concepto de Fibra (Fiber). principios de multitareas

Módulo 3: Planificación de procesadores

Objetivos. Introducción al problema de la planificación: planificación de monoprocesadores y multiprocesadores.

Niveles de planificación: extra largo plazo, largo plazo, mediano plazo y a corto plazo.

Criterios de planificación de los trabajos y de los procesos: política vs. mecanismo.

Administración y gestión de procesos y procesadores: tipos de planificadores. Algoritmos de planificación del procesador.

Algoritmos NON-PREEMPTIVE (sin reemplazo o apropiativos): FCFS (First-Come First-Served), SPF-Shortest Process First (también llamado SPN-Shortest Process Next). Planificación por prioridad.

Algoritmos PREEMPTIVE (con reemplazo en el uso del procesador), Round Robin o torneo cíclico, Menor tiempo restante (SRT Shortest Remaining Time First). Primero el de mayor tasa de respuesta (HRRN).

Planificación con colas de múltiples niveles y realimentación. Planificación con múltiples colas fijas.

Planificación con múltiples colas dinámicas. Planificación de tres niveles. Evaluación y comparación de algoritmos.

Planificación de múltiples procesadores: granularidad, planificación de múltiples procesos y de hilos.

Ejemplos de "scheduler/dispatcher" de sistemas operativos.

Evaluación de desempeño. Detección de cuellos de botellas en el procesador.

Módulo 4: Sincronización y Comunicación entre Procesos

Conceptos de sincronización y comunicación entre procesos.

Problemas concurrentes. Grafos de precedencia. Condiciones de concurrencia (Bernstein).

Especificaciones concurrentes: Fork y Join, Cobegin y coend.

Relaciones entre procesos concurrentes y sus conflictos. Introducción al problema de la región crítica (RC.).

Condición de carrera. Solución de la concurrencia por software y hardware.

Algoritmos de sincronización con espera activa: solución simple, espera ocupada por turnos (alternancia), solución de Peterson, algoritmo de Dekker, algoritmo de Lamport o de la panadería.

Algoritmos sin espera activa: semáforos, monitores.

Mecanismos provistos por el hardware. Cola de espera, Semáforos.

Comunicaciones entre procesos: mensajes, IPC: Inter Process Communication, tipos de sincronizaciones mediante mensajes, modelo productor-consumidor, algunos algoritmos para el modelo productor-consumidor.

Deadlocks (interbloqueo, bloqueo mutuo o abrazo mortal). Condiciones necesarias y suficientes. Tipos de recursos.

Ejemplos de abrazo mortal. Prevención, detección, evasión y recuperación de abrazo mortal.

Métodos de representación: grafos y matrices. Grafo de asignación de recursos. Estrategias para tratar Deadlocks. Conflicto en la comunicación entre procesos

Módulo 5: Administración de Memoria Central

Administración de memoria central (MC). Funciones del administrador de la memoria central. Objetivos de la administración de la MC. Asignación y reasignación de direcciones. Espacio de direcciones lógico y físico.

Técnicas de administración sin swapping: Memoria dedicada (máquina desnuda sin SO.), Asignación contigua simple o monitor residente, asignación particionada simple y variable, paginación pura, segmentación simple, manejo de memoria con buddy system.

Técnicas de administración con swapping (intercambio) o sea memoria virtual: swapping, paginación por demanda o bajo solicitud.

Algoritmos de gestión de memoria virtual, sistemas mixtos: segmentación con paginación por demanda.

Módulo 6: Sistema de Gestión de Entrada / Salida

Administración de la Entrada / Salida (I/O scheduler). Funciones del administrador de E/S. Módulos de E/S y la estructura del módulo de E/S.

Función del módulo. Estructura del módulo de E/S.

Las operaciones del hardware de E/S: operación asincrónica, diferencias de velocidades. Los dispositivos y sus interfaces (el hardware de E/S): dispositivos de E/S. Controlador, adaptador o interface de E/S, procesadores de E/S (IOP), dispositivos externos, almacenamiento intermedio de E/S (Buffering), dispositivos internos.

Técnicas de E/S: E/S programada, E/S por interrupciones, E/S por DMA (Acceso Directo a Memoria).

Principios del software de E/S. Metas del software de E/S. Manejadores de interrupciones (Interrupt handler). Drivers de dispositivos. Pasos y controles en una operación de E/S. Software de E/S independiente del dispositivo.

Software de E/S del espacio del usuario. Software de entrada. Software de salida. Procesadores de E/S y Canales de E/S

Módulo 7: Sistema de archivos y sus directorios

Introducción sistema de gestión de archivos (File System).

Concepto de archivo. Tipos de archivos. Atributos de los archivos.

Sistemas basados en cinta y en disco.

Objetivos y funciones del sistema de gestión de archivos. Conflictos. Sistema básico de archivos. La estructura de la información. Archivos mapeados a memoria. Nombres de archivos. La estructura de un archivo. Estructura interna. Descriptores de archivos.

Operaciones sobre archivos: apertura y cierre, creación, escritura, lectura, rebobinado y borrado.

Catalogación de los archivos en el soporte: Área de datos fijos, área de catálogo y área de datos.

Administración del espacio de almacenamiento: espacio libre, métodos de asignación. Sistemas de directorio: directorio de dispositivos. Operaciones sobre directorios. Estructuras de directorio.

Métodos de acceso: acceso secuencial, acceso directo, otros métodos de acceso. Métodos de implementación del sistema de archivos. Algoritmos para la administración de archivos.

Protección de archivos: nombre, contraseñas, control de acceso.

Módulo 8: Protección y seguridad

Concepto de seguridad y protección. Concepto de política y mecanismo. Política de seguridad. Principios de las políticas de seguridad. Categorías básicas de las políticas de seguridad. Objetivos de la seguridad y la protección de un sistema. Justificación de la seguridad y protección. Niveles de seguridad en informática.

Amenazas a la seguridad. Diseño: principio de los mecanismos. Tipos de seguridad. Supervisión y vigilancia.

Supervisión de riesgos de seguridad por el SO.

Seguridad a través del sistema operativo. Funciones de los sistemas de protección en el sistema operativo.

Seguridad en el kernel.

Dominios de protección: matriz de accesos. Implementación de la matriz de accesos. Cambio de dominio – switch.

Cambio de contenido de la matriz de accesos. Revocación de permisos. Sistemas basados en capacidades.

Seguridad multinivel, autenticación del usuario: validación. Los problemas de la identidad: sus puntos débiles.

Amenazas relacionadas con los programas: caballo de troya, puerta trasera, bomba lógica, desbordamiento de pila y de buffer, virus, gusanos, vulnerabilidad. Política de seguridad.

Seguridad para los datos. Seguridad de datos en bases de datos. Métodos de ocultamiento de los datos.

Algunos problemas en CRIPTOGRAFÍA.

Seguridad en telecomunicaciones o redes de computadoras. Distribución de llaves. Normas y procedimientos en un sistema de seguridad: estrategia de seguridad, plan de contingencia. Auditorías. Mecanismos y políticas de seguridad en sistemas.

Módulo 9: Sistemas distribuidos

Conceptos de sistemas cliente/servidor y sus variantes.

Conceptos de procesamiento distribuido.

Conceptos de sistemas de archivos en sistemas distribuidos.

Conceptos de control de concurrencia en sistemas distribuidos.

Conceptos de memoria compartida distribuida.

Conceptos sobre transacciones distribuidas

Módulo 10: Sistemas de alto rendimiento

Conceptos de procesadores de alta performance.

Conceptos de procesamiento paralelo.

Conceptos de arquitecturas multiprocesadores.

Generación y ajuste de un sistema operativo. Mediciones del sistema y performance.

BIBLIOGRAFÍA RECOMENDADA PARA EL CURSO (EN INGLÉS) 2016

OBRA: Operating Systems Internals and Design Principles (7th Edition)

AUTOR: Stallings, William

EDITORIAL: Prentice Hall

FECHA: 2011

OBRA: Operating Systems Concepts (9th edition)

AUTOR: Silberschatz, J.L. and Galvin P. B.

EDITORIAL: Addison Wesley

FECHA: 2012

BIBLIOGRAFÍA RECOMENDADA PARA CONSULTA (EN INGLÉS)

OBRA: UNIX Internals - A Practical Approach

AUTOR: Steve D Pate

EDITORIAL: Addison Wesley

FECHA: 1996

OBRA: Advanced programming the UNIX environment

AUTOR: Richard Stevens

EDITORIAL: Addison Wesley

FECHA: 2001

OBRA: UNIX network programming Volume 1

AUTOR: Richard Stevens

EDITORIAL: Prentice Hall

FECHA: 1998

BIBLIOGRAFÍA RECOMENDADA PARA CONSULTA (EN CASTELLANO)

OBRA: Notas sobre Sistemas Operativos - Manual del Alumno - 2 tomos

AUTOR: La Cátedra

EDITORIAL: Ghia

FECHA: 2006

OBSER.: Libro de referencia para el seguimiento de las clases

OBRA: Apuntes de Sistemas Operativos Distribuidos

AUTOR: La Cátedra

EDITORIAL: Ghia

FECHA: 2007

OBSER.: Libro de referencia para el seguimiento de las clases

OBRA: Sistemas Operativos Principios de diseño (Desde la Fifth Edition)

AUTOR: Stallings, William

EDITORIAL: Prentice Hall

FECHA: 2006

OBRA: Sistemas Distribuidos – Conceptos y Diseño (Desde la 3° Edition)

AUTOR: George Coulouris / Jean Dollimore / Tim Kindberg

EDITORIAL: Addison Wesley

FECHA: 2001

OBRA: Sistemas Operativos (Desde la 7ta. edición)

AUTOR: Silberschatz, Galvin & Gagne

EDITORIAL: Mc Graw Hill

FECHA: 2006

METODOLOGÍA DE ENSEÑANZA.

El dictado del curso será del tipo explicativo, participativo e informativo, basado en la discusión de los tópicos desarrollados en el transcurso de las diferentes clases mediante su tratamiento teórico y de ejemplos de aplicaciones prácticas.

La introducción de un tema, generalmente es precedida por un diálogo dirigido, con preguntas orientadas hacia el tema a tratar, lo que induce a la participación de todo el grupo.

A partir de esto se desarrolla la exposición teórica con ejercitación práctica en el aula, esta exposición puede ser apoyada por una lectura previa recomendada a los alumnos. Los conceptos impartidos son reforzados y puestos en práctica con los ejercicios propuestos en la Guía de Ejercicios confeccionada por la cátedra. Esa ejercitación permite al alumno confrontar los nuevos conocimientos con los previamente adquiridos y aplicar los conceptos vistos teóricamente, a nuevas situaciones. Algunos ejercicios son presentados, discutidos y resueltos en el aula por el docente

Además la cátedra dispone de una guía de trabajos prácticos para ser desarrollados por los alumnos en forma grupal en el laboratorio de Sistemas Abierto que dispone la cátedra. Cada tema propuesto se verifica en tiempo y forma en el laboratorio observando su correcto funcionamiento en la computadora. Dentro de este ámbito el alumno dispone de atención permanente de docentes para aclarar todas sus consultas. Este estilo de trabajo es abordado durante todas las clases. Cada Trabajo práctico Grupal se deberá defender en forma individual para su aprobación.

Se utilizará material audiovisual cuando las circunstancias así lo requieran.

6.- DESCRIPCIÓN DE LA ACTIVIDAD CURRICULAR

Por parte del Profesor:

- Desarrollo de clases de exposición de temas teóricos Utilizando Pizarrón y en ocasiones con presentaciones por computadora.
- Desarrollo de clases prácticas de resolución de ejercicios de aplicación de los conceptos teóricos.
- Desarrollo de clases prácticas en laboratorio, mostrando ejemplos significativos de la teoría y demostraciones prácticas que deben realizar los alumnos.
- Actualización de contenidos en la página de la cátedra donde se encuentra toda la documentación de la asignatura y medio de comunicación para envío de noticias, material y dar respuesta a los requerimientos de los alumnos (acceso vía e-mail de y a los alumnos).

Por parte de los alumnos:

- Resolución individual de ejercicios de aplicación de la teoría propuestos por el profesor.
- Desarrollo e implementación grupal de una a serie de trabajos Prácticos diseñados especialmente para que el grupo de alumnos los programe.

Material Didáctico:

- Diapositivas Power Point de Clase sobre temas teóricos
- Guías de Trabajos Prácticos y Ejemplos de Resolución de Ejercicios desarrollados por la cátedra
- Procedimientos escritos para diversos procesos.
- Bibliografía básica y avanzada.
- Uso del sitio de la Asignatura.

Envío y atención de mail para consultas

7.- EXPERIENCIAS DE LABORATORIO, TALLER O TRABAJOS DE CAMPO

NOTA: Todos los trabajos prácticos que se realicen en ésta materia serán corregidos en un entorno GNU/Linux o Windows 2012. Para mayor detalle de las versiones utilizadas concurrir al laboratorio 266. En caso de que el alumnado decida realizar los trabajos en otra plataforma, o en otra distribución, deberá tomar las precauciones necesarias para que el producto entregado pueda ser ejecutado en dicho ambiente. En caso de que el producto entregado no cumpla con éstas indicaciones, el trabajo práctico será considerado como no entregado.

Trabajo Práctico Nro. 1:

Scripting

Trabajo Práctico Nro. 2:

Scripting

Trabajo Práctico Nro. 3:

Diagnósticos, Virtualización y Storage en sistemas operativos

Defensa de TP's (Coloquio) y Recuperatorios

Todos los Trabajos prácticos serán probados por los docentes y defendidos por los alumnos en el Laboratorio. En las defensas podrán estar presentes además de los docentes del curso, el jefe de cátedra, y el jefe de trabajos prácticos.

8.- USO DE COMPUTADORAS

En la asignatura se utilizan profusamente computadoras en experiencias de simulación y de programación de Sistemas como complemento práctico de la Teoría. Para ello se recurrirá al Laboratorio específico de Sistemas Operativos (aula 266).

9.- METODOLOGÍA DE EVALUACIÓN

- Esta asignaturas se evaluará de acuerdo a la reglamentación vigente en la Universidad y la que se detalla en el "Reglamento de cursado y aprobación de la materia".
- Se efectuarán dos evaluaciones parciales: El primero al promediar el dictado del curso y el segundo al finalizar el mismo según el Calendario Académico.
- Asimismo y como **condición necesaria para la aprobación del curso** se examinará al alumnado mediante una Guía de Trabajos Prácticos que se desarrollará durante el transcurso del mismo, además de las exposiciones orales que efectuarán los alumnos sobre los T.P., cuestionarios o problemas teóricos planteados.
- Además se requiere una asistencia a clase no inferior al 75%, se hace un seguimiento del trabajo realizado por cada integrante en cada clase.

El conjunto formado por los Trabajos Prácticos y las evaluaciones parciales serán el instrumento para medir el rendimiento y la aprobación de la cursada.

Aprobación y cursada

A los fines de la aprobación de la materia, se considera "la última nota obtenida" en cada uno de los exámenes rendidos (en primera instancia ó recuperatorios).

- a) Por régimen de promoción, sin examen final, se considera la materia **aprobada**, cuando la calificación es igual o superior a 7 (siete) a través de exámenes parciales y recuperatorios, en las fechas indicadas en el cronograma.
- b) Si el alumno no llena los requisitos para promover (calificación superior o igual a 4 pero inferior a 7 puntos), queda en condición de **cursada**. Para su aprobación definitiva tiene que rendir posteriormente un examen final. La validez de la cursada será de 5 turnos consecutivos de examen final. Dichos turnos serán contados a partir del turno inmediato siguiente al periodo de cursada.
- c) El alumno que tenga 1 (un) aplazo en las evaluaciones y/o recuperatorios, y haya estado presente en todas las instancias evaluativas, pierde la materia y se considera **desaprobado**.
- d) Aquel alumno que tenga al menos 1 (un) examen cuya evaluación final sea ausente (considerando parcial y recuperatorio), se considera **ausente**.

Régimen de Trabajos Prácticos

- 1- En fecha de entrega del TP, se hará una corrección grupal de ejercicios en forma arbitraria para cada grupo.
- 2- La NO presentación del TP o estar incompleto en la fecha propuesta significa su desaprobación en primera instancia (es para todos los integrantes del grupo).
- 3- La realización de cada TP será grupal pero su evaluación individual a través de un examen escrito u oral en todos los trabajos prácticos.
- 4- El alumno que no apruebe la evaluación del TP o NO se entregó en fecha establecida, tendrá una nueva fecha que es a la semana siguiente del establecido inicialmente en el cronograma.
- 5- El alumno que desapruebe 2(dos) TP en segundas instancias (por ausencia, estar incorrecto, y /o incompleto o no responder correctamente el coloquio) desaprobará la materia.

Examen Final

- Los alumnos pueden rendir examen final bajo dos modalidades **regular** o **libre**.
- Para rendir examen como **regular** deberá tener la materia cursada y no haberse operado el vencimiento de la misma.
- Deberán rendir como regular los que obtengan entre cuatro y seis en los parciales o sus recuperatorios.
- Para rendir examen como **libre** tendrán que ajustarse a la reglamentación vigente.
- La mesa examinadora considerará válidas las inscripciones que consten en las actas proporcionadas por la oficina de alumnos.
- Cada alumno rendirá el final con el programa vigente.

Condiciones para rendir EXAMEN LIBRE

Por Resolución N° 142 del H.C.S., se autoriza a rendir “**exámenes libres**” de todas las asignaturas, a los alumnos de las tres Carreras pertenecientes al Departamento de Ingeniería e Investigaciones Tecnológicas.

Todos los alumnos estarán en condiciones de rendir exámenes libres, siempre y cuando hayan aprobado las materias correlativas correspondientes.

Dicha instancia examinadora se deberá llevar a cabo en una de las fechas de convocatoria a exámenes finales.

Para mayor detalle sobre la forma de rendir exámenes libres y los requerimientos a cumplir antes de presentarse en la llamada correspondiente ver la sección REGLAMENTO DE CURSADA LIBRE DE LA CÁTEDRA SISTEMAS DE COMPUTACIÓN II (PLAN 1997) / SISTEMAS OPERATIVOS (PLAN 2009) más adelante en el presente documento

CALENDARIO DE ACTIVIDADES (modalidad presencial)

PLANIFICACIÓN DOCENTE PARA EL AÑO 2016

DURACIÓN DE CADA CURSO:

- Teórica: Aprox. 8 clases de 4 horas. (32 horas)
- Práctica: Aprox. 9 clases de 4 horas. (36 horas)
- Laboratorio: Aprox. 13 clases de 4 horas. (52 horas)

HORARIO: Según el fijado para cada curso (turno mañana 8 a 12 / turno noche 19 a 23)

CRONOGRAMA DE ACTIVIDADES DE LA PLANIFICACIÓN POR CURSO

Clase 1: Introductoria-Práctica. Presupuesto de tiempo: 4 Hs.

Fecha: Comienzo del ciclo lectivo 2016- (16/08/2016)

Objetivos:

- Definir la metodología para el futuro desarrollo del curso y dar los lineamientos introductorios al curso. Explicar la metodología de evaluación de TPs y exámenes parciales
- Introducción al sistema operativo GNU/Linux

Tipo de conocimiento:

- Práctico

Evaluación del Módulo:

- Primer parcial

MÓDULO 1: Presupuesto de tiempo: 2 Hs.

Objetivos:

- Que el alumno incorpore un enfoque introductorio sobre los Sistemas Operativos, sus interfaces, los servicios que brinda, su funcionamiento y conozca la terminología básica y conceptual de los S.O. y sus ambientes de trabajo.

Tipo de conocimiento:

- Teórico

Evaluación del Módulo:

- Primer parcial

MÓDULO 2: Presupuesto de tiempo: 2 Hs.

Objetivos:

- Que el alumno adquiera los conocimientos sobre las distintas modalidades de procesamiento e incorpore los conceptos fundamentales sobre organización del ambiente de ejecución.

Tipo de conocimiento:

- Teórico

Evaluación del Módulo:

- Primer parcial

PRÁCTICO 1: Presupuesto de tiempo: 8 Hs.

Objetivos:

- Que el alumno incorpore los conocimientos para la codificación de scripts básicos y con las herramientas awk y sed

Tipo de conocimiento:

- Práctica de laboratorio

Evaluación del Módulo:

- Coloquio del TP

EVALUACIÓN PRÁCTICO 1: Presupuesto de tiempo: 1 Hs.

Objetivos:

- Evaluar a los alumnos sobre los conocimientos de GNU/Linux adquiridos durante el trabajo práctico 1

MÓDULO 3: Presupuesto de tiempo: 8 Hs.**Objetivos:**

- Que el alumno se familiarice con los conceptos y los medios de la planificación del procesador y de los procesos, en especial en el largo, mediano y corto plazo.

Tipo de conocimiento:

- Teórico y práctico

Evaluación del Módulo:

- Primer parcial

MÓDULO 4: Presupuesto de tiempo: 10 Hs.**Objetivos:**

- Que el alumno integre los conceptos fundamentales sobre los recursos compartidos, sincronización y comunicación entre procesos.

Tipo de conocimiento:

- Teórico y práctico

Evaluación del Módulo:

- Primer parcial

PRÁCTICO 2: Presupuesto de tiempo: 6 Hs.**Objetivos:**

- Que el alumno incorpore los conocimientos en PowerSell necesarios para los trabajos prácticos

Tipo de conocimiento:

- Práctica de laboratorio

Evaluación del Módulo:

- Coloquio del TP

EVALUACIÓN PRÁCTICO 2: Presupuesto de tiempo: 1 Hs.**Objetivos:**

- Evaluar a los alumnos sobre los conocimientos de GNU/Linux adquiridos durante el trabajo práctico 2

EVALUACIÓN 1: Presupuesto de tiempo: 4 Hs.**Objetivos:**

- Evaluar a los alumnos sobre los conocimientos teóricos y prácticos.

MÓDULO 5: Presupuesto de tiempo: 10 Hs.**Objetivos:**

- Que el alumno concrete los conceptos sobre la administración de la Memoria Central, en especial las particiones y los conceptos de asignación, paginación y segmentación.

Tipo de conocimiento:

- Teórico y práctico

Evaluación del Módulo:

- Segundo parcial

MÓDULO 6: Presupuesto de tiempo: 6 Hs.**Objetivos:**

- Que el alumno incorpore los conceptos sobre la administración de los dispositivos de Entrada - Salida.

Tipo de conocimiento:

- Teórico y práctico

Evaluación del Módulo:

- Segundo parcial

MÓDULO 7: Presupuesto de tiempo: 6 Hs.**Objetivos:**

- Que el alumno conozca los métodos de acceso para el almacenamiento y la recuperación de la información en los soportes como también la administración de la misma.

Tipo de conocimiento:

- Teórico y práctico

Evaluación del Módulo:

- Tercer parcial

MÓDULO 8: Presupuesto de tiempo: 1 Hs.**Objetivos:**

- Que el alumno conozca los fundamentos y los conceptos sobre el manejo de la protección y la seguridad de un centro de cómputo y el S.O.

Tipo de conocimiento:

- Teórico

Evaluación del Módulo:

- Segundo parcial

PRÁCTICO 3: Presupuesto de tiempo: 10 Hs.**Objetivos:**

- Que el alumno incorpore los conocimientos necesarios para realizar el diagnóstico en sistemas operativos

Tipo de conocimiento:

- Práctica de laboratorio

Evaluación del Módulo:

- Coloquio del TP

EVALUACIÓN PRÁCTICO 3: Presupuesto de tiempo: 1 Hs.**Objetivos:**

- Evaluar a los alumnos sobre los conocimientos de GNU/Linux adquiridos durante el trabajo práctico 3

EVALUACIÓN 2: Presupuesto de tiempo: 4 Hs.**Objetivos:**

- Evaluar a los alumnos sobre los conocimientos teóricos y prácticos.

MÓDULO 9: Presupuesto de tiempo: 1 Hs.**Objetivos:**

- Que el alumno adquiera los conceptos básicos sobre métrica de sistemas

Tipo de conocimiento:

- Teórico

Evaluación del Módulo:

- Sin evaluación

MÓDULO 10: Presupuesto de tiempo: 6 Hs.**Objetivos:**

- Que el alumno adquiera los conceptos básicos sobre los sistemas operativos distribuidos, sus problemáticas y la forma de implementar las soluciones

Tipo de conocimiento:

- Teórico

Evaluación del Módulo:

- Sin evaluación

RECUPERACIÓN 1: Presupuesto de tiempo: 4 Hs.**Objetivos:**

- Que los alumnos tengan la posibilidad de recuperar los exámenes que tengan aplazados o con notas menores a 7

REGLAMENTO DE PROMOCIÓN y NORMAS DE LA CÁTEDRA

(Cursada normal)

1. La aprobación de la asignatura *Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)* en el período lectivo **2016** se basará en:
 - 1.1. La Normativa vigente en la Universidad,
 - 1.2. Las Normas Básicas de la Cátedra *Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)* que se detallan a continuación en tanto reglen aspectos no normados por los elementos anteriores.
 - 1.3. El Reglamento Interno de la *Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)* que se detalla a continuación, en este documento.

NORMAS BÁSICAS DE LA CÁTEDRA

1. El dictado de la materia se dividirá en aproximadamente 32 clases teóricas y clases prácticas, según calendario adjunto, las que incluirán clases teóricas, práctica, de evaluación y de recuperación.
2. En las clases de contenido teórico se desarrollarán los temas teóricos establecidos en el programa analítico adjunto. En las clases prácticas los alumnos, orientados por los docentes a cargo de las mismas, resolverán problemas y ejercicios de aplicación de los temas vistos en clase y los de la presente guía de trabajos prácticos.
3. La aprobación de los trabajos prácticos (Firma de Libreta), se obtendrá a través de:
 - 3.1. La presentación y aprobación de los trabajos prácticos, según lo detallado en la presente.
 - 3.2. La aprobación de, al menos, dos exámenes parciales en las fechas y condiciones establecidas en el calendario adjunto y en el régimen de cursado y aprobación de ambas asignaturas siguiente:

Régimen de cursado y aprobación de la asignatura *Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)*

1. ASISTENCIAS:

- Se requiere una asistencia a clases no inferior al 75% (setenta y cinco %) tanto para la modalidad presencia como la semipresencial. El incumplimiento de este requisito coloca al alumno en condición de "ausente" o "desaprobado".

2.- RÉGIMEN DE PROMOCIÓN POR EXÁMENES PARCIALES Y RECUPERATORIOS:

- La asignatura se aprueba por régimen de promoción por exámenes parciales y/o recuperatorios.
- Para esta asignatura el cursado es cuatrimestral y habrá 2 (dos) evaluaciones parciales: uno por bimestre. Las instancias recuperatorias serán 1 (una). En esta asignatura se entiende "**ausente**" al alumno que no posea 2 (dos) evaluación parciales rendidas y no haya entregado más de 2 (dos) trabajos prácticos.
- Los exámenes parciales (y sus recuperatorios) se entenderán "**aprobados**" cuando la calificación asignada, en una escala de 0 a 10 puntos, resulte superior o igual a 7 (siete) puntos.
- La asignatura se entenderá "**aprobada**" cuando se aprueben todos los exámenes parciales (en primera instancia o por recuperatorio). La calificación asignada al examen recuperatorio (cualquiera sea el resultado), **anula y reemplaza**, a todos los efectos, la obtenida en el examen parcial que se recupera. La calificación final se calculará como promedio de los exámenes parciales o el último recuperatorio de cada parcial, rendidos y aprobados. Es importante mencionar que en caso de tener una nota entre 4 (cauto) y 6 (seis) en el parcial y sacarse un 2 (dos) en el recuperatorio, la materia se considerará como "**desaprobada**".
- De esta manera la calificación final necesaria para que la asignatura resulte "**aprobada**" deberá ser superior o igual a 7 (siete) puntos, pero no se podrán poseer notas menores a 7 (siete).

3. RÉGIMEN NO PROMOCIONADO DE PARCIALES Y SUS RECUPERATORIOS:

- En la asignatura, los exámenes parciales (y sus recuperatorios) calificados con 3 (tres) o menos puntos se entenderán "**aplazados**" y podrán ser recuperados.
- Si el alumno al finalizar la cursada tiene algún parcial (o recuperatorio) y/o trabajo práctico calificados con aplazo, se considerará "**aplazada**" la materia y deberá ser recursada en otro cuatrimestre.
- Los exámenes parciales calificados con 4 (cuatro), 5 (cinco) ó 6 (seis) puntos, se entenderán "**aprobados**" y podrán ser recuperados (en caso de que el alumno desee la promoción de la materia), pero no se podrá con estas notas conseguir la calificación final de "**Aprobado**", teniendo en este caso la condición final de "**Cursada**".
- La asignatura con calificación final, calculada como promedio de los exámenes parciales (o los recuperatorios correspondientes) rendidos y no aplazados, de 4 (cuatro), 5 (cinco) o 6 (seis) puntos, se entenderán "**cursada**" y podrá ser aprobada a través de un "**examen final**".
- La calificación necesaria para aprobar el "**examen final**" será de 4 o más puntos

4.- LA VALIDEZ DE LA ASIGNATURA "CURSADA"

- La validez de la asignatura “**cursada**” se rige bajo las normas de la facultad, por lo que se deberá consultar con las autoridades pertinentes.

5.- PREREQUISITO CONDICIONANTE PARA RENDIR LOS EXAMENES PARCIALES:

- Esta asignatura requiere que el alumno tenga el porcentaje de asistencia correspondiente para poder rendir los exámenes parciales y recuperatorios

REGLAMENTO INTERNO DE LA CÁTEDRA SISTEMAS DE COMPUTACIÓN II (PLAN 1997) / SISTEMAS OPERATIVOS (PLAN 2009)

1. OBJETIVO:

- Dar las bases normativas por las que se regirá el funcionamiento y el desarrollo operativo de la cátedra.

2. ALCANCES:

- El presente Reglamento **NO EXCLUYE** a la reglamentación vigente, sino todo lo contrario, pretende complementarla para lograr las metas operativas propuestas para cada curso en particular.

3. CONTENIDO:

a) **DE LOS PROGRAMAS:** El contenido es el indicado en el programa analítico de la materia.

b) **DEL CRONOGRAMA DE ACTIVIDADES:** Se ajustará de acuerdo al presupuesto de tiempo previsto en la planificación docente y se formalizará el primer día de clase en cada curso en particular.

c) **DE LA ASISTENCIA:** Es de recalcar que la asistencia, en el caso específico de ésta materia, juega un rol importante debido al intenso ritmo que se impartirá al dictado de las clases teóricas, por lo tanto se recomienda al alumno concurrir a dichas clases, siendo de su exclusiva responsabilidad cumplir con este requisito. Pero se aplicará el punto 1 del Régimen de cursado y aprobación de las asignaturas *Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)*.

d) **DEL HORARIO:** En el inicio de la clase, la puntualidad es importante a los fines de constituir un ambiente ordenado. Se recomienda al alumnado el cumplimiento de este requerimiento. En particular también se recomienda la permanencia dentro del aula mientras se desarrollan las clases.

e) **DE LAS CLASES TEÓRICAS:** El Docente y sus Ayudantes dictarán la materia tratando de seguir la secuencia estricta de los módulos y la Planificación propuesta. El desarrollo tendrá un carácter ampliamente comunicativo que permita la participación del alumnado. El método a aplicar será explicativo-inductivo-deductivo. Tanto el docente o sus Ayudantes evacuarán las dudas que surjan durante el dictado de las clases o de los T.P.

f) **DE LOS TRABAJOS PRÁCTICOS:** Los alumnos confeccionarán una serie de Trabajos Prácticos (TPs.), para ello se dispondrá de una Guía de T.P. (adjunta al presente documento). Cada Guía deberá ser completada en la fecha establecida por la Cátedra o el docente a cargo del curso. La totalidad de las guías formarán una “**carpeta de T.P.**”

Los T.P. se dividirán en dos categorías: 1) Optativos y 2) Obligatorios.

Cada guía deberá ser entregada, por el alumno o el grupo de alumnos, **en la fecha planificada a los efectos de ser corregida.**

OBSERVACIÓN: LOS T.P. NO ENTREGADOS EN FECHA SE CONSIDERAN NO APROBADOS.

La guía corregida por la cátedra será devuelta con las observaciones correspondientes para que los alumnos procedan a rectificar lo solicitado. Una vez cumplimentado por los alumnos, en el plazo fijado, los T.P. serán entregados a la cátedra para su aprobación. La cátedra firmará la aprobación parcial de cada Guía y devolverá el original para que cada alumno pueda disponer de una constancia de la aprobación, la que integrará una “carpeta de TPs”. Todos los TPs originales aprobados formarán una “carpeta de T.P. originales” que deberá ser presentada al final de la cursada.

g) **DE LA PRESENTACIÓN DE LOS T.P.:** La presentación se deberá realizar en dos soportes: Hojas de papel y medio electrónico (ver reglamento de entrega más adelante en esta guía).

La presentación que se realiza en hojas de papel, deberá ser normalizada en papel A-4, o carta (no se aceptarán entregas en papel oficio), y los contenidos impresos se ajustarán a las “**Normas para la presentación escrita de los Trabajos Prácticos**” que figuran en la guía de T.P.

h) **DE LA EVALUACIÓN DE LOS T.P.:** Todos los puntos se evaluarán mediante las consideraciones en particular de cada ítem siguiente:

- Desarrollo por temas (extensión).
- Contenidos (Calidad y en el caso de programas: funcionamiento).
- Criterios.
- Síntesis.
- Definiciones (acotaciones).
- Alcances.
- Investigación Bibliográfica.
- Presentación.

El conjunto de notas dará como resultado la aprobación o desaprobación del T.P. en particular.

i) **DE LA REGULARIZACIÓN DE LA MATERIA:** Para la firma de la Libreta, el alumno deberá presentar:

- La Libreta Universitaria.

- Haber aprobado los T.P. realizados durante el curso ya sean grupales o individuales.
- Tener todos los parciales aprobados y cumplir con lo dispuesto en el Régimen de cursado y aprobación de la asignatura Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009).
- Ser alumno regular.
- Realizar el "POSTEST" que propondrá la Cátedra y la encuesta.

j) **DE LAS EVALUACIONES DURANTE EL CURSO:** Habrá dos evaluaciones parciales durante el curso. El docente fijará con cada curso fecha de cada uno de esos parciales y la del recuperatorio. Habrá un recuperatorio en el que podrá rendirse uno de los dos parciales según lo especificado en el régimen de aprobación de la materia.

Los T.P. grupales serán expuestos en el pizarrón o en una reunión grupal con el Jefe de Trabajos Prácticos o docente del curso, por cada integrante del grupo a los fines de examinar su participación en el desarrollo del T.P. y que dará lugar a una evaluación de cada presentación individual. Además de considerar una nota única por cada T.P. grupal.

k) **DE LAS EVALUACIONES FINALES:** Los mismos pueden ser Teórico-prácticos y en forma escrita y/u oral, según lo aconsejen las circunstancias.

La examinación se hará a través de un Tribunal Examinador. Para poder rendir el examen final los alumnos deberán tener regularizada la materia y la correlatividades respectivas de esta materia.

REGLAMENTO DE CURSADA LIBRE DE LA CÁTEDRA SISTEMAS DE COMPUTACIÓN II (PLAN 1997) / SISTEMAS OPERATIVOS (PLAN 2009)

1. OBJETIVO:

- Dar las bases normativas por las que se implementará la aprobación de la materia a través de exámenes libres para conseguir la aprobación de la cátedra.

2. ALCANCES:

- El presente Reglamento **NO EXCLUYE** a la reglamentación vigente, sino todo lo contrario, pretende complementarla para lograr las metas operativas propuestas para cada curso en particular.

3. CONTENIDO DEL EXAMEN LIBRE:

a) **DE LOS TRABAJOS PRÁCTICOS:** El alumno que desee rendir la materia en condición de libre deberá efectuar **TODOS** los trabajos prácticos que la cátedra haya dispuesto para el cuatrimestre en curso vigente en la guía de trabajos prácticos que suministra la materia, confeccionándolos y teniéndolos que presentar con 15 días de anticipación a la fecha de rendir el examen libre. La vigencia de los trabajos prácticos para el examen libre será desde el comienzo del cuatrimestre correspondiente al que se quiere rendir el examen libre hasta el comienzo del próximo cuatrimestre. Esto quiere decir que por ejemplo si se desea rendir el examen libre correspondiente a la cursada del segundo cuatrimestre 2016, la llamada a examen en la que el alumno podrá presentarse son las correspondientes a julio de 2016, en caso de querer rendir el examen libre correspondiente a la cursada del segundo cuatrimestre de 2016, las llamadas en las que podrá presentarse son las de diciembre de 2016, y marzo de 2017 y en todas aquellas fechas de examen intermedias que disponga la Universidad.

b) **DE LA EVALUACIÓN DE LOS T.P.:** Los trabajos prácticos entregados por el alumno que rinde el examen libre serán evaluados en los 15 días que hay hasta la fecha del examen final por los docentes de la cátedra y en caso de estar bien, el alumno deberá rendir un coloquio como primera parte del examen final. Los puntos se evaluarán mediante las consideraciones en particular de cada ítem siguiente:

- Desarrollo por temas (extensión).
- Contenidos (Calidad y en el caso de programas: funcionamiento).
- Criterios.
- Síntesis.
- Creatividad.
- Definiciones (acotaciones).
- Alcances.
- Investigación Bibliográfica.
- Presentación.

El conjunto de notas dará como resultado la aprobación o desaprobación de los trabajos prácticos.

c) **DE LA EVALUACIÓN FINAL:** En caso de aprobar los trabajos prácticos (tanto la presentación, como el coloquio), el alumno deberá rendir un examen final para la condición de libre, que tendrá una primera parte práctica escrita (conteniendo ejercicios tanto de la práctica de clases como de la práctica de laboratorio). En caso de aprobar dicho examen deberá pasar un examen teórico con carácter oral que incluirá todo el contenido de la materia que se indica en el PROGRAMA ANÁLITICO de la materia.

“Certifico que el presente programa de estudios de la asignatura *Sistemas de Computación II (Plan 1997)* / *Sistemas Operativos (Plan 2009)* es el vigente para el segundo cuatrimestre del ciclo lectivo 2016, guarda consistencia con los contenidos mínimos del plan de estudios y se encuentra convenientemente actualizado”

Fabio E. Rivalta
Jefe de Cátedra
Universidad Nacional de La Matanza
15/08/2016

COMENTARIOS PREVIOS

Estos comentarios la opinión de la cátedra respecto de la metodología de estudio, conocimientos previos y la infraestructura necesaria; para el correcto cursado de la materia. Como estos aspectos no dependen exclusivamente del esfuerzo de la cátedra de S.O., es posible que algunas de estas opiniones no sean realidad aun.

RESPECTO DE LOS CONOCIMIENTOS PREVIOS NECESARIOS

Los conocimientos previos que se requieren del alumnado para comprender la temática de la materia y estudiarla con cierto grado de profundidad, básicamente se agrupan en los cuatro siguientes campos: hardware, software, estructuras de datos y algoritmia y dentro de ellos específicamente los siguientes temas:

Hardware:

- Automatas finitos,
- Álgebra de códigos y álgebra binaria,
- Conceptos de arquitecturas computacionales,
- Direccionamientos del procesador,
- Memoria principal, rango y resolución,
- Periféricos, controladores, canales, interfases, dispositivos,
- Distintos soportes de información, grabación y recuperación,
- Registros de la CPU, ALU, unidad de control,
- Interrupciones.

Software:

- Instrucciones primitivas, lenguajes de máquinas, niveles de lenguajes,
- Instrucciones comunes y privilegiadas, macros
- Concepto de trabajo, paso de trabajo, proceso
- Concepto de traductores y editores de enlace
- Concepto de editores de texto
- Concepto de programa, rutina, autorutina y corutina
- Concepto de declaraciones, declarativas, variables locales y globales.
- Procedimientos, recursividad.

Estructura de datos:

- Registros, archivos, punteros, operadores,
- Arreglos, estructuras, unión, apuntadores o punteros,
- Proposiciones y asignaciones,
- Expresiones,
- Vector, tabla, cola, lista, pila, árboles

Algoritmia:

- Lenguaje c, Pascal o Modula 2
- Lenguajes orientados a objetos (c++)

El alumno que entienda no conocer alguno de los puntos antes citados debería consultar bibliografía adecuada, con el propósito de adquirir el o los conocimientos en cuestión. Este punto es fundamental ya que estos conceptos son esenciales para la comprensión de la materia.

RESPECTO DEL TRABAJO A REALIZAR

- En cuanto a los esfuerzos que el alumno debe dedicar normalmente a esta materia comprenden dos aspectos: uno teórico y otro práctico.
- El teórico consiste en el seguimiento de las clases, que en general se prevé para el cuatrimestre, de cuatro horas reloj por clase con un total de 28 clases para el curso, al cual se deberá agregar un tiempo adicional de lectura de texto (ver bibliografía recomendada), con igual carga horaria. Se recomienda en especial prestar atención a las lecturas de los textos, sobre todo las propuestas en idioma inglés.
- El aspecto práctico queda determinada por la experiencia, con que cuenta cada alumno, en programación mediante lenguajes estructurados y su modalidad de trabajo en laboratorios (cerrados, colaboración con grupos o individual), ya sea hogareña o en los puestos de trabajo cotidiano. De todas formas consideramos que el mínimo esfuerzo en la faz práctica requiere de dos a tres horas diarias de dedicación frente al computador y el mismo tiempo de lectura o estudio por clase.
- Las clases prácticas serán desarrolladas por los Docentes auxiliares en el horario determinado para estas tareas.

RESPECTO DEL LABORATORIO (AULA 266)

- Con respecto al laboratorio, consideramos interesante que los alumnos utilicen el equipamiento ordenadamente solicitando los respectivos turnos con el debido tiempo. Que se dispongan a trabajar no más de tres a cuatro alumnos por máquina.
- El laboratorio deberá estar en condiciones de uso antes y luego de que los alumnos de la cátedra accedieron a la instalación. Respetando la normativa establecida para su correcto uso.
- El trabajo en el laboratorio deberá estar organizado por turnos en el que los alumnos se presentan al mismo, habiendo previamente elaborado la práctica o ejercicios. Recomendamos especialmente no perder tiempo (y hacer perder tiempo), con improvisaciones sin previa fundamentación. Los Ayudantes o instructores estarán disponibles para resolver las consultas específicas que le formulen los alumnos sobre los trabajos y ejercicios propuestos.
- La experiencia nos dice que durante el tiempo previsto, los alumnos no completan las tareas solicitadas de acuerdo al planeamiento propuesto para el desarrollo del curso y solo lo hacen a último momento, por lo que los horarios del laboratorio se satura, entonces también es recomendable que los alumnos utilicen racionalmente al laboratorio dentro de la amplitud horaria que les fuera asignada y no padezcan los efectos de la demanda de último momento.

NORMAS PARA LA PRESENTACIÓN ESCRITA DE LOS T.P.

Introducción

Este documento tiene por objetivo dar algunas guías y sugerencias que ayudarán a los alumnos a producir sus trabajos prácticos con un alto grado de cuidado, precisión y elegancia. También pretendemos homogeneizar la presentación de los trabajos solicitados durante el curso. Los siguientes lineamientos son aplicables a todos los trabajos prácticos propuestos.

Las normativas que a continuación se enumeran tienen carácter de **complementario**, es decir que, bajo ningún punto de vista se deberá usar el presente documento como refutación o reemplazo de las normas vigentes establecidas por la Universidad. De la misma forma, las normas establecidas en el documento de planificación de la Cátedra, tienen carácter de marco regulatorio en vista a las normativas para este curso.

La presente documentación está sujeta a cambio, es responsabilidad del alumno notificarse de las alteraciones producidas. Para tal fin, las versiones que surgieren durante el transcurso del año lectivo estarán disponibles en las siguientes locaciones:

Directorio de la Cátedra, Server del Laboratorio de la Universidad.

Lineamientos Generales

Para la correcta presentación de cualquier trabajo se deberán seguir las siguientes pautas generales:

a) Carátula o Tapa de Encuadernación

Al comienzo del trabajo deberá figurar una carátula (normalizada según documento provisto por la cátedra) debidamente completada.

b) Escrito, formato de contenidos

Para la presentación del trabajo deberán utilizarse dos medios de almacenamiento distintos. El primero es en papel tamaño A4, o carta (no oficio). El segundo en formato digital, referirse al apartado "**Reglamento de entrega y reentrega de Trabajos prácticos**" para más información. Todos los escritos y representaciones gráficas deberán ser impresos y el disco deberá contener los archivos de datos que generan las impresiones y los programas correspondientes. De ninguna manera serán aceptados los trabajos manuscritos, salvo en las revisiones **informales**.

NOTA: el equipo de trabajo debe garantizar que el material entregado, en cualquier tipo de medio magnético u óptico, está libre de virus y defectos. La no observancia de esta normativa implicara un grave daño para la evaluación del material entregado.

La tipografía a utilizar deberá ser clara, de fácil lectura y compatible con sistemas OCR (Optical Character Recognition), sugiriéndose: letra de imprenta, en cualquiera de sus estilos (preferentemente Arial o Roman), no aceptándose letras cursivas, ni góticas. El tamaño de letra a utilizar para el cuerpo de texto, deberá ser de 10 a 12 puntos. El cuerpo de texto no podrá ser todo en mayúsculas. Todos los comienzos de párrafo deberán tener una sangría de ocho espacios al margen izquierdo y justificados a ambos márgenes del documento.

Deberá incluirse un pie y encabezado de páginas en todas las hojas, menos en carátulas. El pie de página deberá contener el número de página como mínimo, en cursiva y con el prefijo "Página". El encabezado

deberá contener el título del trabajo y el nombre del equipo, en un tamaño de letra que no difiera con el del texto y en negritas.

c) Centrado del Documento

El material o contenido debe estar dispuesto simétricamente con referencia al centro del texto escrito y no equidistante de los bordes, el margen izquierdo deberá ser mayor al derecho para permitir la encuadernación. Los dibujos, tablas, gráficos y demás objetos incrustados, también se colocarán con referencia al centro del texto y no de la hoja en sí. El centrado vertical deberá ser el conveniente para la correcta impresión y visualización de los encabezados y pies de páginas.

d) Márgenes y sangrías

De acuerdo a las Normas IRAM para el formato A-4 corresponde unos 20 mm como mínimo para el margen izquierdo, y 20 mm para el derecho, 25mm para el borde inferior de la hoja y desde el octavo renglón del borde superior. Es aconsejable respetar esta normativa, sin embargo es posible usar estos parámetros en función de la correcta visualización del material. Las sangrías deberán ser respetadas a lo largo de todo el documento, utilizando la capacidad de tabulación automática de algunos procesadores de texto.

e) Separación de Palabras, micro justificación

Si es necesario realizar la división de palabras, debe hacerse siempre entre dos sílabas. Los números, fórmulas, fechas, y nombres propios no se dividirán. La micro justificación de los caracteres deberá ser fijada en forma automática.

f) Alineación de Enumeración de Títulos y Subtítulos

Los encabezamientos deben cumplir un doble fin: poner título a la sección o grupo de párrafos y facilitar la consulta rápida. Para ello se establece la siguiente estructura de los encabezamientos:

1. Título principal o Nivel 1
 - 1.1. Subtítulo o Sección Nivel 2
 - 1.1.1. Enumeración Temática o Nivel 3
 - 1.1.2.
 - 1.1.3.
 - 1.2.
 - 1.2.1.
 - 1.2.1.1. Categorización de contenidos o Nivel 4
 - 1.2.1.2.
 - 1.2.2.
 - 1.3.
 - 2.

Cuando deben alinearse números arábigos o romanos deberá procederse de la siguiente forma: (alineando la cifra final y no la primera)

| | | |
|-----|-------|-----|
| 9 | (i) | I |
| 12 | (ii) | II |
| 111 | (iii) | III |

Las carátulas, el índice y la Introducción deberán numerarse en formato romano, exceptuando de ello a la carátula principal. De la misma forma se deben numerar los capítulos y secciones del documento. Para tal fin debe crearse los estilos correspondiente en la galería de estilos del procesador de textos, e incluirlos como marcadores para la generación del índice de contenidos.

Las carátulas no llevarán encabezamiento ni pie de página.

g) Espaciado Vertical

Todo el texto del documento debe ser escrito a un espacio. El título inicial se coloca a cinco espacios del margen superior o del encabezamiento y luego a tres espacios se comienza a escribir. Los encabezamientos importantes se colocarán a tres espacios antes y a tres después; los demás títulos tienen tres espacios antes y dos después.

Los capítulos deberán indicar en su encabezamiento, en las páginas pares el título del trabajo, y en las impares, el capítulo y el tema del cual trata. En cada pie de página figurará el número de página correspondiente, numeradas en forma consecutiva con números arábigos. Tanto el encabezamiento como el pie de página, deberán estar separados del área de texto por una línea trazada desde el margen izquierdo al margen derecho.

h) Títulos, Subtítulos y Secciones Destacadas

Los títulos principales o de Nivel 0, tendrán que figurar centralizados, subrayados, con letra itálica y resaltada, con un tamaño que oscile entre los 20 y 30 puntos. Los subtítulos deberán ir numerados, con el número de capítulo, un punto y el número de sección, se ubicarán desde el margen izquierdo con sangría explicada en el punto f), subrayados y en letra resaltada. Para éstos se utilizara un tamaño de letra entre los 12 y 16 puntos.

Los títulos de párrafos figurarán sangrados desde el margen izquierdo, con el mismo tamaño de letra que se utiliza en el cuerpo principal, subrayados y en negrita. Deberán estar numerados con el número de capítulo, el número de sección y el número de párrafo, separados por puntos.

Para resaltar palabras o frases dentro del cuerpo de texto se podrán utilizar tanto el resaltado como la letra itálica, siempre que se respeten el tamaño y el tipo de letra utilizado en el cuerpo de texto.

Cuando se deba enumerar un contenido, éste tendrá que estar tabulado y para la enumeración se utilizarán números arábigos, separados por un guión del texto.

Los cuadros o gráficos (en lo posible), serán escritos en el texto directamente.

i) Del Índice General

El título del mismo deberá figurar como cualquier título del resto del trabajo. Cada capítulo figurará al margen izquierdo, con su correspondiente número, separado por un guión del nombre, subrayado y en letra resaltada, y no deberán llevar número de página.

Los subtítulos figurarán en un primer grado de tabulación, con el número formado de la misma manera en que se colocaron en el resto de la obra, y a continuación el nombre en letra resaltada. En caso que un tema no quepa en un solo renglón, su continuación, deberá estar alineada con la primer letra del nombre del tema, además, el número de página (si correspondiese), en que se encuentra, deberá figurar en el último renglón y alineado al margen derecho.

Los títulos de párrafos figurarán en un segundo grado de tabulación, con el mismo número que tienen en el cuerpo del texto, con letra resaltada y cumpliendo los términos que se detallan en el párrafo anterior.

Los apéndices figurarán como capítulos independientes, pero no llevarán numeración en sus subtítulos y títulos de párrafos. De la misma forma, figurará la bibliografía.

j) Los Apéndices o Anexos

Los apéndices contendrán información relacionada con el tema principal del trabajo. En ellos figurarán todos aquellos datos y documentos que complementen el texto, y que figurando en el cuerpo principal harían dificultosa su lectura.

Los apéndices deberán enumerarse alfabéticamente, los títulos, subtítulos y títulos de párrafos cumplirán las mismas normas que las del cuerpo principal con la excepción de la numeración.

Cada uno de los apéndices en lo posible, deberá presentarse con un breve encabezamiento, o si el alumno lo prefiere, una carátula en la que figure, el número de apéndice y el tema tratado.

La numeración de las páginas de un apéndice será correlativa a la del resto del trabajo.

k) La Bibliografía

En ella deberá figurar todo aquel material que se haya consultado o estudiado para la confección del documento del trabajo práctico. Se detallará en letra resaltada el título de la obra, seguido por el / los autores en letra normal y en los renglones siguientes, en forma tabulada, la editorial, año de edición y demás datos complementarios. La bibliografía siempre se coloca al final del documento.

También se deben incluir los URL de los sitios de Internet consultados.

l) Encuadernación

Los trabajos serán presentados adecuadamente encuadernados o encarpetados. Este puede ser con carpetas tipo plásticas, con carpeta transparente y lomo de plástico, con anillado o tipo libro con tapas ad hoc. El método de encuadernación utilizado deberá contener los aditamentos necesarios para la contención del medio magnético u óptico asociado al trabajo.

m) Identificación:

Todas las hojas presentadas, en trabajos de teoría o prácticas, (salvo la carátula) deberán contener en el Encabezado o Pie de página el nombre del autor o grupo, curso, fecha de presentación y nombre del trabajo. El tamaño de letra a utilizar para el encabezado, deberá ser de 8 a 10 puntos.

NO SE ACEPTARÁN HOJAS SUELTAS.

Los Archivos en los soportes ópticos – magnéticos se identificarán de la siguiente forma:

<Apellido del Alumno (o grupo)_Siglas de la Universidad_Curso_año_Título de la entrega>

Ejemplo: "PEREZ UNLAM Com35 2016 monografía sobre Device Drivers"

Todas las presentaciones de trabajos teóricos en medios ópticos o magnéticos deberán tener una etiqueta rotulada con el Nombre del Trabajo o Número de presentación, Apellido y Nombre del Alumno (o todos los integrantes del grupo), Universidad Curso, año, título.

La presentación de Monografías, deberán estar contenidas en un solo archivo.

Consideraciones especiales

A continuación se detallan el uso de algunas características especiales para la presentación de los trabajos prácticos. Solo deberán ser tomadas en cuenta en las situaciones convenientes.

a) Uso de campos especiales

Algunos procesadores de texto permiten la inclusión de campos especiales, como ser campos de datos o formularios. Siempre que sea conveniente, la inclusión de estos campos deberá estar documentada en un apartado. La experiencia indica que "sobrecargar" un documento con estas características puede ser contraproducente, sin embargo a los efectos de comprensión del tema expuesto es posible que se presente la necesidad de uso de estas facilidades.

Se deberá utilizar un utilitario editor standard en el mercado para la inclusión de campos especiales como ser, archivos de sonidos, gráficos, u otros datos que puedan adjuntarse al documento. Es aconsejable el uso de herramientas pertenecientes a un mismo paquete.

b) Galería de estilos

Cada documento deberá tener una galería de estilos utilizados. El objetivo de esta facilidad es mantener en forma uniforme el formato del documento. Es aconsejable que cada equipo genere un standard a utilizar en todos los documentos que entregue. Cada uno de los equipos deberá generar un archivo de descripción de estilos (.dot en el caso de Microsoft Word) y adjuntarlo a cada uno de los documentos.

c) Software sugerido

La siguiente lista de software tiene carácter de sugerencia debido a que son estándares del mercado en la actualidad. Cualquier otro aplicativo que el equipo de trabajo desee utilizar deberá ser provisto a la cátedra en su debido tiempo y forma.

- Procesador de Texto, Microsoft Word 97 o superior
- Aplicativo de Presentaciones, Microsoft PowerPoint 97 o superior
- Planilla de cálculos, Microsoft Excel 97 o superior
- Bases de Datos, cualquier motor que respete la norma DBASE III
- Generador de Páginas HTML, Microsoft FrontPage
- Aplicación para graficación de Esquemas, Visio Tech 4.0 o superior

Es aconsejable el uso de herramientas desarrolladas con fines específicos, como es el procesador de texto LATEX.

d) Formato del medio óptico o magnético

Se deberá generar la siguiente estructura para la entrega del material. La etiqueta del medio deberá contener el nombre del grupo y el título del trabajo entregado, la etiqueta que se genera con el formateo del medio debe contener el número del grupo y la clave de la entrega (nn-clave), según las siguientes normas:

Nn: número del equipo de trabajo

Clave: <F|R><ff> donde F: Final, P: Revisión, ff: fecha empaquetada

La estructura de directorios deberá ser la siguiente:

```
\<Raiz>
  Documentación
    Manual de Uso
    Desarrollo Temático
    Documentación de Soporte
    Fuentes
  Ejecutables
    Aplicativo
      Versión 1
      Versión 2
      Versión n
    Utilidades
  Editores
  Bibliografía
```

Reglamento particular de Entregas

Esta guía de Trabajos Prácticos consta de cuatro trabajos todos de carácter grupal, que son obligatorios para la aprobación de la parte práctica de la materia, y que junto a la aprobación de la parte teórica (parciales / parcialitos), formarán la nota final de la materia Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)

Introducción

Objetivos

1. Generar un marco de trabajo para la puesta en marcha de diversos aspectos teóricos de la materia Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)
2. Desarrollar actividades que faciliten la adquisición de nuevos conocimientos en el ambiente de Sistemas Operativos

Alcances

Todos los módulos de la materia presentados en la guía introductoria.

Estas Normativas Complementarias son aplicables a todas las normas y procedimientos establecidos en la Universidad para el desenvolvimiento de un curso práctico.

Son aplicables todas las normas y procedimientos internos de la Cátedra Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)

Rol del Equipo de Trabajo

Para los trabajos prácticos grupales, los alumnos deberán formar grupos de trabajo con un mínimo de una persona (no es recomendable, pero admitido), y un máximo de 5 (cinco) alumnos. Todos los integrantes de un grupo tienen que pertenecer a la misma comisión, no estando permitido formar grupos con integrantes de más de una comisión.

A diferencia de años anteriores, los grupos formados tendrán validez solamente por un trabajo práctico, lo que permite a los alumnos a cambiar de grupo entre un trabajo y otro. Para tal fin, el alumno deberá presentar al inicio del trabajo práctico un formulario indicando cual será el grupo con el que trabajará para dicho trabajo práctico.

Cabe aclarar que en ningún caso se permitirá cambiar de grupo en el medio de un TP., y que en caso de tener que reentregar / recuperar un trabajo práctico luego de un cambio de grupo, lo deberá realizar con el grupo original. Esto quiere decir que el grupo conformado para un determinado trabajo práctico es inmutable, y deberá ser concluido por los alumnos que continúen la cursada dentro de las fechas reglamentadas.

Rol de Líder de Equipo

Una vez formado el equipo de trabajo, deberán elegir un líder de grupo. Los mecanismos para la selección serán de entera responsabilidad del equipo en cuestión. Una vez fijado el miembro líder, este mantendrá su rol hasta finalizado el práctico. Este miembro es el responsable de generar el "Reglamento de Convivencia" que debe ser observado por el resto del grupo. El desempeño de este rol no tiene ningún tipo de impacto en vista a las evaluaciones grupales. Las responsabilidades "extras" de este rol serán fijadas oportunamente en cada curso.

Rol de Supervisor

Este rol será llevado adelante por un docente responsable de la práctica de la materia, asignado al grupo en cuestión. El objetivo de este rol es suministrar un lineamiento general en el desarrollo de los temas asignados a un equipo determinado.

Rol de Revisor

La cátedra nombrará un docente en carácter de revisor para la evaluación individual de cada trabajo entregado.

Mesa de Examen

El Jefe de la cátedra o el revisor presidirá la evaluación final con todo el equipo docente que haya intervenido en el desarrollo de los trabajos.

Mecánica de los Trabajos Prácticos.

Las normas redactadas a continuación tienen por objetivo fijar un marco de trabajo que permita un desarrollo ordenado de las prácticas.

Modalidad de Trabajo

La modalidad de trabajo para la práctica será la de "trabajo en equipo supervisado", modalidad que imprime un ritmo dinámico a las clases prácticas. La responsabilidad de desarrollo de la práctica recae en el equipo en general y por igual a todos sus miembros, sin considerar roles especiales de los mismos (líder de grupo). El equipo de trabajo retiene en su esfera de actividades todas las cuestiones relacionadas a la administración del tiempo y el planeamiento para la calidad. Toda problemática, que no sea estrictamente relacionada con la temática a desarrollar, será discutida y resuelta en forma interna en el equipo de trabajo.

Evaluación del Trabajo Práctico.

Ámbito:

Es aplicable a todos los cursos de la cátedra de Sistemas de Computación II (Plan 1997) / Sistemas Operativos (Plan 2009)

Evaluación de Contenidos

La evaluación de contenidos se realizará sobre la base de lo expuesto en la guía de la materia.

Evaluación de Cohesión

La evaluación de cohesión tiene como objetivo ponderar el trabajo de cada uno de los individuos como miembros de un equipo.

Evaluación de Exposición y Defensa

La evaluación de exposición y defensa tiene como objetivo ponderar al equipo y a sus miembros en la claridad y solidez con que exponen los temas desarrollados. Así mismo, se evaluará la solidez de las demostraciones para respaldar las teorías expuestas.

Evaluación de Capacidad de Crítica

La evaluación de capacidad de crítica tiene como objetivo ponderar al equipo y a sus miembros en la capacidad de crítica frente a la exposición de otro equipo o dentro del suyo propio. Solo serán permitidas las críticas al modelo presentado, bajo ningún punto de vista será permitida una crítica a la mecánica o constitución de otro equipo de trabajo. Esta evaluación se aplicará en los cursos o grupos que decidan exponer sus trabajos a la crítica de otros grupos.

Evaluación y Calificación Finales

El conjunto de evaluaciones parciales, arriba expuesto, será fuente de información valiosa para la estructuración de una calificación y referencias finales.

Actividades Prácticas

Entregas de Trabajos

Durante el ciclo lectivo, las actividades prácticas propuestas resultarán en un “Entregable”

Se observarán dos tipos de entrega:

Entregas de Revisión

Pueden ser efectuadas en cualquier momento, sin importar el orden de secuencia de los temas a desarrollar. Deberán observarse todas las normas que estén vigentes para las entregas. La frecuencia de estas entregas es la estipulada en cada TP., no pudiéndose realizar más de una por semana. Tiene carácter obligatorio la primera entrega de revisión al promediar la mitad del periodo previsto para el tema en cuestión. El resto de las entregas es de carácter opcional. Las evaluaciones de los trabajos entregados, en carácter de revisión, son tomadas en cuenta para la evaluación del trabajo final.

Entregas Finales

Son de carácter obligatorio, en tiempo y forma. La no entrega de un trabajo final genera su desaprobación automática. Se deberán observar las fechas y horarios de entrega, los trabajos que superen ese lapso de tiempo serán recibidos en calidad de recuperatorios.

En caso de no entregarse en término, como penalidad, se asignarán nuevos trabajos prácticos de mayor dificultad. Todos los Trabajos Prácticos deberán estar aprobados en la fecha prevista para la firma, en caso de no hacerlo, se asignará un trabajo extra de Recuperatorio que el alumno deberá completar y entregar funcionando en la fecha que se determine para cada caso. Esta entrega se efectuará durante los días fijados para ello y constará, además de los Trabajos Prácticos adeudados, de un examen el recuperatorio integrador de la materia. Cabe aclarar que la dificultad del trabajo práctico adicional será mucho mayor que la de todos los Trabajos Prácticos asignados durante la cursada.

Los Trabajos Prácticos, deberán ser entregados en los medios indicados en cada trabajo, respetando para todos ellos el reglamento de presentación de TP, como ser un mismo tipo de letra, formato de papel, etc.

En las carátulas de presentación de todos los Trabajos Prácticos, se deberá consignar, número de trabajo práctico y versión de entrega y los nombres, apellidos y matrículas o legajos de todos los integrantes del grupo. En caso de entregar medios de almacenamiento, en la etiqueta de los mismos, deberán figurar los mismos datos que en la carátula. Si se tratase de una entrega complementaria, se deberá anexar una descripción de las modificaciones efectuadas con respecto a la entrega anterior. No se aceptarán entregas que no cumplan las condiciones solicitadas, esto quiere decir, que si la presentación se debe realizar en disquete, y papel, no se aceptará que se entregue uno, y no el otro (Ej. el disquete solo, o la carpeta sola).

En los trabajos entregados en medios magnéticos se deberá tener especial cuidado en la presencia de virus informáticos y posibles errores en el medio de almacenamiento, en caso de existir no se considerará entregado dicho trabajo y el mismo deberá ser RECUPERADO. También se debe tener en cuenta, los posibles errores de lectura de las

unidades magnéticas, para esto se recomienda grabar el medio en una unidad, y probar copiarlo desde otra.

Los materiales entregados pasan a ser propiedad de la cátedra, por lo cual se pide tener especial cuidado en no dejar en los medios a entregar información de otras materias, o importantes para el alumnado, y que no tiene que ver con la materia, y/o trabajo práctico entregado.

También se debe tener en cuenta que los docentes de la cátedra, se toman el trabajo de comparar el código fuente de cada uno de los trabajos con los de otros grupos, comisiones e incluso años, por lo tanto se les recomienda no copiarse los trabajos, puesto que en caso de existir fraude, se perderá automáticamente el trabajo, y se les incorporarán a los grupos intervinientes en dicho fraude, sendos trabajos prácticos, los que deberán ser entregados junto con el trabajo falseado, efectuado nuevamente. Cuando se indica "grupos intervinientes", se considera tanto el grupo que copió el trabajo como el grupo al que le pertenece el trabajo copiado. Por esto se sugiere tener especial atención con los trabajos que se dejan dentro de las cuentas de los servidores de la Universidad, y sobre todo con las passwords seleccionadas, el cambio periódico de las mismas y su confidencialidad.

Defensa del Trabajo Entregado

La defensa de un trabajo tiene carácter obligatorio y consiste en responder todas las cuestiones formuladas por los docentes que evalúan los trabajos entregados. La defensa del trabajo se realizará en el horario y fecha preestablecido por los docentes responsables de estas actividades.

Exposición y Defensa de Trabajos Prácticos

Ámbito:

Esta modalidad es válida para aquellos cursos que decidan que cada grupo exponga su trabajo a la crítica de los restantes grupos.

Criticas a Trabajos

La crítica a los trabajos de otros equipos se regirá por la dinámica grupal en una clase práctica. Todas las críticas deberán estar debidamente fundamentadas y posteriormente escritas, pues deben figurar en el informe final de la presentación.

Defensa del Trabajo

La defensa de un trabajo consiste en escuchar y aceptar las críticas de otros grupos de pares que enriquecerán la posterior refutación y/o reconocimiento de las propuestas de cambio. En la dinámica grupal las refutaciones pueden ser orales, siendo necesario transcribirlas posteriormente en el informe final de la presentación. La defensa del trabajo se realizará en el horario y fecha preestablecido por los docentes responsables de estas actividades.

Publicaciones

Trabajos a Publicar

Todos los trabajos realizados están sujetos a publicación en la biblioteca de la Universidad. El objetivo es poner a disposición de los demás alumnos los trabajos realizados, como fuente de información y referencia futura.

Derechos

Se deja constancia que la Cátedra se compromete a no hacer usufructo comercial de los desarrollos realizados por los alumnos. Sin embargo, se reserva el derecho de utilizar dichos desarrollos para futuros prácticos que permitan enriquecer los resultados que se obtengan. Al finalizar el presente cuatrimestre se podrán seleccionar los mejores trabajos para tomar como base de prácticos de futuros cursos como también su posible utilización (sin lucro) por parte de la Universidad, con la debida mención de sus autores.

Condiciones para la Publicación

Si bien todos los trabajos pueden ser publicados, al finalizar el año se seleccionarán los trabajos de mejor calidad. Los factores a tomar en cuenta son los mismos para las evaluaciones. Otra condición de publicación del material desarrollado es la aprobación por parte del director de la cátedra y de una mesa examinadora especialmente formada para tal fin.

Mecanismos de Publicación

Los trabajos seleccionados deberán ser adecuadamente encuadernados por el equipo de trabajo y se suministrarán dos copias idénticas de los mismos. Las dos copias serán donadas a la biblioteca de la Universidad y se permitirá el préstamo en calidad de material de consulta. Todas las normas para entregas se aplican a las copias de publicación.

Calendario de Entregables

A determinar oportunamente por los docentes responsables de las cátedras. En general se informará el calendario en cada curso.

Descripción de los Trabajos Prácticos

Trabajo Práctico Nro. 1:

Scripts con PowerShell en Windows

Trabajo Práctico Nro. 2:

Scripts con bash, awk, sed y señales en sistema operativo GNU/Linux

Trabajo Práctico Nro. 3:

Diagnósticos, Virtualización y Storage en sistemas operativos

Defensa de TP's (Coloquio) y recuperatorio de parciales:

Los recuperatorios de los parciales y trabajos prácticos serán coordinados por el grupo docente de cada una de las comisiones con los alumnos durante el transcurso de la cursada.

Notas importantes:

- Si un trabajo no es entregado en la fecha correspondiente, se contará con un período de 7 (siete) días corridos para su entrega (denominada fecha tardía de entrega – que se encuentra aclarada en cada trabajo práctico-). Una vez transcurrido el plazo no se aceptará la entrega hasta la fecha de recuperación. Es importante mencionar que no todos los trabajos prácticos tienen fecha de entrega tardía.
- Para los trabajos prácticos que tengan fechas de entregas graduales, se considerará cada entrega parcial como un trabajo práctico por separado. De forma que si un trabajo práctico está dividido en dos entregas, serán como dos trabajos independientes pero que conformarán una sola nota; y para la nota final del trabajo práctico se tomará el promedio de las notas de las dos entregas.
- No se podrá entregar un trabajo práctico sin que se encuentre entregado el anterior, y aprobados todos los trabajos prácticos previos al anterior. Ej.: Si se quiere entregar el trabajo práctico número 3, se deberá tener presentado para su evaluación el trabajo práctico 2, y aprobado el trabajo práctico 1. Ver más detalle en el Reglamento de entrega

Reglamento de entrega y reentrega de Trabajos prácticos

Todos los trabajos prácticos tienen el mismo formato de entrega o reentregas que es descrito a continuación, y que en caso de no ser cumplido no será considerado como entregado el trabajo práctico.

Entrega escrita:

Por cada entrega o reentrega de trabajo práctico se deberá entregar una carpeta o folio conteniendo:

- **Una (1) ó Dos (2) carátulas** (dependiendo de lo que indique el docente del curso) utilizando para tal fin la plantilla disponible en el sitio web de la cátedra debidamente completada.
- Para poder realizar la entrega de cualquier trabajo práctico se deberá cumplir con las siguientes condiciones para cada uno de ellos:

| TP | Entrega en fecha | Entrega tardía | TPs entregados | TPs Aprobados |
|----|----------------------------------|----------------------------------|----------------|---------------|
| 1 | Sin ejercicios de entrega tardía | Con ejercicios de entrega tardía | -- | -- |
| 2 | Sin ejercicios de entrega tardía | Con ejercicios de entrega tardía | TP 1 | -- |
| 3 | TP Resuelto | N/A | TP 1 y TP 2 | TP 1 |

Entrega digital:

Los trabajos prácticos deberán entregarse mediante la opción [Entrega de TPs](#) del sitio web de la cátedra (<http://www.sisop.com.ar>). Es importante tener en cuenta que la entrega la deberá realizar sólo uno de los integrantes de cada grupo. Queda a consideración de cada equipo de trabajo quién será el que realice las entregas.

El sistema aceptará un único archivo comprimido por cada entrega, con extensiones **zip, gzip, gz o tgz**.

El avance de las correcciones, feedback de los profesores y notas podrán visualizarlos todos los integrantes del grupo (sin importar quién hizo la entrega) desde la opción de menú [Mis Notas](#).

Guía orientativa para el uso de GNU/Linux

- **Descripción:** A continuación se detallan una serie de preguntas y ejercicios orientados al uso de una terminal de caracteres de un ambiente multiusuario y multitarea basado en GNU/Linux. La intención buscada con esta guía es que aquellos alumnos que no estén familiarizados con una terminal de caracteres ni con la familia de sistemas operativos GNU/Linux, puedan aprender las principales características y poder realizar los trabajos posteriores con un mejor conocimiento del entorno en el que se deben realizar.
- **Nota:** Si bien este no es un trabajo práctico de entrega obligatoria, aquel alumno/a que crea conveniente realizarlo y desee consultar a los docentes o ayudantes de la materia sobre su contenido podrá hacerlo en cualquiera de las clases prácticas. Desde el cuerpo docente recomendamos a todos aquellos alumnos que nunca hayan trabajado en un ambiente de este tipo realicen esta guía y consulten a los docentes sobre los temas aprendidos y los no comprendidos
- **Formato de entrega:** sin entrega o con entrega de consulta optativa
- **Preguntas:** A continuación se detallan todas las preguntas y ejercicios que deberán ser resueltos. Tenga en cuenta que salvo en los momentos que indica que debe estar sesionado como **root**, en el resto de los ejercicios debe estar conectado como usuario común. (TIP: se le recomienda que primero realice todo el trabajo, anotando los resultados en papel, y a mano, y luego lo pase con el editor vi).

1. INTRODUCCIÓN

- 1.1. ¿Qué es la cuenta de superusuario (root) y para qué se utiliza?
- 1.2. Ingresar al sistema como superusuario (root), y realizar los siguientes pasos (éste punto no puede ser realizado en el laboratorio 266):
 - 1.2.1. adduser <apellido> (reemplazar <apellido> por el suyo).
 - 1.2.2. passwd <apellido> (Ingrese una contraseña (password) a su elección).
 - 1.2.3. logout
- 1.3. Indique claramente qué efectuaron estos comandos, e indique qué archivo/s fueron modificados (Dentro del directorio /etc) **TIP:** Utilice lo siguiente: "ls -lt /etc | more".
- 1.4. Luego ejecute "cat /etc/passwd | more" y haga lo mismo con los otros archivos que se modificaron. Analice y comente lo visto
- 1.5. ¿En qué directorio se encuentran los comandos utilizados en los puntos 1.2.1, 1.2.2, 1.2.3, 1.3, y 1.4?

2. AYUDA

- 2.1. INFO: Info es un programa para leer documentación. Este se compone de una estructura del tipo árbol, dividido en nodos de información. Cada nodo describe un específico tópico con un determinado nivel de detalle.
 - 2.1.1. Ingrese a info y responda:
 - 2.1.1.1. ¿Cómo se llama el nodo raíz de Info?
 - 2.1.1.2. Ubique el cursor en la línea (* cp:) y presione ENTER.
 - 2.1.1.3. ¿Qué sucedió?
 - 2.1.1.4. ¿Cómo se llama este nodo?
 - 2.1.1.5. ¿Cuál es el próximo nodo?
 - 2.1.1.6. ¿Cómo puedo moverme al próximo nodo?
 - 2.1.1.7. ¿Cómo puedo moverme al nodo anterior?
 - 2.1.2. Presione la tecla 'u'.
 - 2.1.2.1. ¿Qué sucedió?
 - 2.1.2.2. ¿En qué nodo se encuentra?
 - 2.1.3. Repita el punto 2.1.2. hasta que llegue a la raíz de Info.
 - 2.1.3.1. ¿Con qué tecla puedo volver directamente a este nodo?
 - 2.1.3.2. ¿Cuál es el método directo para acceder al nodo cp? (tip: sin desplazar el cursor).
 - 2.1.4. ¿Cómo puedo buscar una palabra clave dentro de un nodo?
 - 2.1.5. ¿Cómo puedo buscar la siguiente palabra clave, buscada anteriormente?
 - 2.1.6. ¿Cómo puedo salir de Info? - salga.
- 2.2. MAN: man es un programa que formatea y muestra la páginas del manual.
 - 2.2.1. ¿Cuál es la diferencia entre man e info?
 - 2.2.2. ¿Cómo puedo ver la información de un determinado comando?
 - 2.2.3. ¿Cómo puedo buscar una palabra clave dentro de la página del manual?
 - 2.2.4. ¿Cómo puedo salir?
 - 2.2.5. ¿Cómo hago para buscar una palabra clave determinada en todas las páginas del manual?
 - 2.2.6. ¿Qué es lo sucede al realizar lo siguiente?
 - 2.2.6.1. man
 - 2.2.6.2. man man
 - 2.2.6.3. man cp
 - 2.2.6.4. man printf

- 2.2.6.5. man fprintf
- 2.2.6.6. man sprintf
- 2.2.6.7. man cd
- 2.2.6.8. man 3 printf
- 2.2.7. Del punto anterior, responder:
 - 2.2.7.1. Al invocar man junto con fprintf y sprintf muestra la misma página. ¿Por qué no muestra la misma página al invocarlo con printf?. (TIP: vea el punto 3.2.6.2).
 - 2.2.7.2. ¿Cómo puedo invocar al man para ver directamente la función printf del lenguaje C?
- 2.3. HELP: help es la ayuda que ofrece el shell de GNU/LINUX para utilizar sus comandos.
 - 2.3.1. ¿Cuál es la diferencia entre help e info?.
 - 2.3.2. ¿Cuál es la diferencia entre help y man?.
 - 2.3.3. ¿Qué sucede al invocar al help?.
 - 2.3.4. ¿Cómo puedo ver la información de un determinado comando?.
- 2.4. whereis
 - 2.4.1. ¿Qué sucede al utilizar el comando whereis cd?
 - 2.4.2. ¿Qué es la información que se muestra por pantalla al ejecutar el punto anterior?
 - 2.4.3. ¿Qué ocurre si se ejecuta whereis * sobre un directorio? (TIP: si no pasa nada, intente nuevamente pero primero ejecute cd /bin)
 - 2.4.4. ¿Cuál es la diferencia entre whereis y find?
- 2.5. whatis
 - 2.5.1. ¿Qué sucede al utilizar el comando whatis cd?
 - 2.5.2. Si el resultado del punto anterior fue la leyenda "cd: nothing appropriate", utilice el comando /usr/sbin/makewhatis, y responda los siguientes puntos:
 - 2.5.2.1. ¿Qué realizó la sentencia anterior?
 - 2.5.2.2. Reintente el punto anterior.
 - 2.5.3. Cambie al directorio /bin, y ejecute el comando whatis * ¿Qué ocurrió?
 - 2.5.4. Utilice el comando apropos passwd y whatis passwd. Enumere las diferencias encontradas en el resultado de cada uno de los comandos.

3. TECLADO / TERMINALES

- 3.1. ¿Qué sucede si tecleo cat /e <tab> p <tab>? (donde tab es la tecla tabulación). Presione <tab> nuevamente ¿Qué pasó ahora?
- 3.2. ¿Qué sucede si tecleo cat /e <tab> pas <tab>?
- 3.3. En este punto analizaremos las distintas terminales que hay en un sistema GNU/Linux. Ejecute los siguientes comandos e indique cuál fue el resultado:
 - 3.3.1. who
 - 3.3.2. Presione la tecla <alt>, y sin soltarla presione cualquiera de las teclas de función. En la pantalla debería aparecer el login del sistema, de lo contrario, ejecute el paso nuevamente presionando otra tecla de función. Si ya tiene el login del sistema vuelva a conectarse.
 - 3.3.3. Ejecute nuevamente el comando who. ¿Qué diferencias encuentra con la primera vez que lo ejecutó?
 - 3.3.4. Ejecute el comando who am i ¿qué muestra?, ¿Qué diferencias tiene con el comando ejecutado en el punto anterior?
 - 3.3.5. Repita el paso 3.3.2 y el 3.3.3 hasta que no encuentre ninguna sesión para abrir.
 - 3.3.6. Una vez terminado el punto anterior, Ud. se encontrará sesionado en el sistema como mínimo seis veces. Lo que acaba de hacer es abrir seis terminales virtuales (que podrían ser usadas por distintos usuarios, con diferentes perfiles), en la misma máquina. Así como existen terminales virtuales dentro del mismo equipo, si Ud. cuenta con una red, o con terminales tipo serie, podría abrir tantas sesiones de trabajo como Ud. quiera o necesite. Investigue e indique cómo se denominan los distintos tipos de terminales, y cuáles son los archivos que las representan (tip: busque en el directorio /dev).

4. DIRECTORIOS

- 4.1. ¿Para qué se usa el comando cd?. Ejecute las siguientes variantes de cd e indique cuál fue el resultado obtenido:
 - 4.1.1. cd /
 - 4.1.2. cd
 - 4.1.3. cd /etc
 - 4.1.4. cd..
 - 4.1.5. cd ..
- 4.2. Bash sobre directorios:
 - 4.2.1. ¿Cuál/es son las diferencias entre el path absoluto y el path relativo?
 - 4.2.2. ¿Qué es lo que realizan las siguientes operaciones? (tip: si no encuentra la diferencia primero haga cd /, y luego vuelva a intentar)
 - 4.2.2.1. cd ~
 - 4.2.2.2. cd -
 - 4.2.3. ¿Cuál es la diferencia entre cd .. y cd --?
- 4.3. Operaciones con directorios:
 - 4.3.1. ¿Con qué comando se puede crear un directorio?.
 - 4.3.2. ¿Con qué comando se puede borrar un directorio?.
 - 4.3.3. ¿Qué sucede si el directorio no está vacío?.
 - 4.3.4. ¿Cómo puedo salvar la situación anterior? (Sin borrar uno a uno los archivos existentes).
- 4.4. ¿Qué significa la expresión ./ cuando se utiliza delante de un archivo? ¿Para qué sirve?
- 4.5. ¿Cómo puede moverse entre directorios sin utilizar el PATH completo?
- 4.6. ¿Cuál es el contenido de los siguientes directorios que confirman la estructura de cualquier sistema operativo GNU/Linux:?
 - 4.6.1. /boot
 - 4.6.2. /dev
 - 4.6.3. /bin
 - 4.6.4. /etc
 - 4.6.5. /usr
 - 4.6.6. /sbin
 - 4.6.7. /root
 - 4.6.8. /etc/rc.d (y todos los que están adentro)
 - 4.6.9. /proc
 - 4.6.10. /mnt
 - 4.6.11. /usr/bin
 - 4.6.12. /usr/sbin
 - 4.6.13. /var
 - 4.6.14. /usr/man (y todos los que están adentro)
 - 4.6.15. /opt
 - 4.6.16. /tmp

5. ARCHIVOS

- 5.1. ¿Qué hacen los siguientes comandos?
 - 5.1.1. cp
 - 5.1.2. mv
 - 5.1.3. rm
 - 5.1.4. rcp
 - 5.1.5. rsh
 - 5.1.6. scp
 - 5.1.7. ssh
- 5.2. Para cada comando del punto anterior realice un ejemplo, e indique qué realizó.
- 5.3. ¿Con qué comando puedo concatenar el contenido de dos archivos?
 - 5.3.1. ¿Se puede usar ese comando para otra cosa?.
- 5.4. Haga un ls -l /dev
 - 5.4.1. ¿Qué significa el primer carácter?
 - 5.4.2. ¿Cuáles son todos los posibles valores que puede contener ese campo y que significa cada uno?
- 5.5. ¿Para qué sirve el comando touch? ¿qué utilidad le encuentra?

6. PERMISOS

- 6.1. Teniendo en cuenta el ls -l anterior, ¿indique que son los siguientes 9 caracteres? (sin considerar el primero sobre el que ya respondió anteriormente)
- 6.2. ¿qué significa cada caracter? ¿cómo están agrupados?
- 6.3. ¿Cómo se asignan los permisos? (detalle los comandos).
- 6.4. ¿Qué son el owner, y el group de un archivo?. ¿Se pueden cambiar?.
- 6.5. Intente cambiar los permisos de un archivo perteneciente al root (sesionado como usuario). Explique qué sucedió.
- 6.6. Explique la forma de cambiar los permisos con valores en octal.
- 6.7. ¿Cuál es el significado de los permisos en los directorios (se debe indicar que indica una r, una w, y una x)?

7. FILTROS

- 7.1. ¿Cuál es la diferencia de los comandos more, less y cat?. De un ejemplo de cada uno.
- 7.2. ¿Cuál es la diferencia entre tail y head?.
- 7.3. ¿Para qué sirve el comando wc y que indican los parámetros -c -l -w? ¿Proponga ejemplos de uso?
- 7.4. ¿Qué es lo que realiza el comando uniq?.
- 7.5. ¿Qué es lo que realiza el comando grep?.
- 7.5.1. ¿Para qué sirve?
- 7.5.2. ¿Qué hace la siguiente línea?: `grep root /etc/passwd`
- 7.5.3. ¿Qué diferencias encuentra entre la ejecución de los siguientes comandos?:
 - 7.5.3.1. `grep r /etc/passwd`
 - 7.5.3.2. `grep ^r /etc/passwd`
 - 7.5.3.3. `grep r$ /etc/passwd`

8. VI

- 8.1. Ejecute la siguiente instrucción: `vi $HOME/prueba.txt` ¿Qué sucedió?. Ahora ejecute todos los pasos detallados a continuación.
 - 8.1.1. Escriba la siguiente frase: "Este es el archivo prueba.txt de <nombre y apellido>"
 - 8.1.2. ¿Qué tuvo que hacer para poder escribir la frase?
 - 8.1.3. Guarde el archivo, y salga del editor. ¿Qué comando utilizó?
 - 8.1.4. Ingrese nuevamente al archivo.
 - 8.1.5. Incorpore al inicio del archivo el siguiente párrafo (los acentos puede ser evitados):
"Sistemas Operativos
Comisión de los días <día de cursada>
Trabajo Práctico 1
Alumno: <su nombre aquí>
Matrícula: <su matrícula aquí>
Documento: <su documento aquí>"
 - 8.1.6. Describa todos los pasos que tuvo que realizar.
 - 8.1.7. Guarde el archivo y continúe la edición. ¿Qué comandos utilizó?
 - 8.1.8. Borre la línea de "Matrícula". Indique por lo menos dos formas de realizarlo.
 - 8.1.9. Invierta el orden de las líneas "Comisión y TP". No está permitido rescribirlas. ¿Qué comandos utilizó?
 - 8.1.10. Ubíquese en la línea 2 (dos) del archivo. No está permitido usar las teclas del cursor, ni el mouse. ¿Qué comando utilizó?
 - 8.1.11. Marque para copiar las líneas 2, 3, y 4 (todas juntas, no de a una a la vez). ¿Cómo lo realizó?
 - 8.1.12. Ubíquese al final del archivo (sin usar las teclas del cursor), y pegue dos veces el contenido del buffer. ¿Qué comando usó?
 - 8.1.13. Deshaga uno de los copiados. No está permitido borrar línea por línea, ni carácter a carácter. ¿Qué comando usó?
 - 8.1.14. ¿Cómo busco la palabra "Documento"? ¿Cómo busco la segunda ocurrencia de una palabra?
 - 8.1.15. ¿Cómo puedo reemplazar la palabra "Documento" por "Documento:" (sin borrar, o realizar el reemplazo a mano)?
 - 8.1.16. Guarde el archivo y salga.
 - 8.1.17. Ejecutar "vi buscar_reemplazar" e introducir el texto:
1/5/2006 ----- listo
1/5/2007 ----- listo
1/5/2008 ----- listo
1/5/2009 ----- listo
1/5/2010 ----- listo
1/5/2011 ----- listo
1/5/2012 ----- listo
1/5/2013 ----- listo
1/5/2014 ----- listo
1/5/2015 ----- listo
1/5/2016 ----- No listo
 - 8.1.18. Ejecutar `:%s/V3V/Marzo/g` ¿Que paso al ejecutar esto?
 - 8.1.19. Si observa el resultado de lo anterior, el cambio fue erróneo, modifique la sentencia para que funcione correctamente.
 - 8.1.20. Modifique la fecha para que en lugar del 1 sea el 15. Indique que comandos uso para realizarlo.
 - 8.1.21. ¿Indique si existe alguna forma de hacer un buscar y reemplazar pero que antes de realizar la sustitución pregunte?.

9. DISCO

- 9.1. ¿Para qué se utiliza el comando mount?. ¿Todos los usuarios lo pueden ejecutar el comando con algunos o todos los parámetros?. En caso de que su respuesta sea negativa, ¿indique cuál /es si?.
- 9.2. Transfiera el archivo a un disquete (el mismo que utilizará para entregar el trabajo práctico, ya que éste archivo es parte de la entrega).
 - 9.2.1. Indique al menos dos formas de realizarlo.
- 9.3. ¿Recordó desmontar el disquete en todas las oportunidades que lo uso, y antes de retirarlo verdad?.
 - 9.3.1. ¿Qué problemas se pueden generar por no realizarlo?.
 - 9.3.2. Repita el punto 1 de éste trabajo práctico, creando un usuario cualquiera (si Ud. se encuentra en el Lab 266, no puede continuar con éste punto).
 - 9.3.3. Cambie de terminal virtual a otra, si se encuentra sesionada salga, e ingrese con el usuario creado en el punto anterior.
 - 9.3.4. Intente desmontar el disquete. ¿Pudo?. Si su respuesta es negativa lo mismo pasará en el laboratorio si Ud. se retira de trabajar sin desmontar la disquetera, y el próximo usuario la quiere utilizar. Por ese motivo en el laboratorio al hacer el logout del sistema se ejecuta un script que verifica si la disquetera está montada. En caso de estarlo, la desmonta, y además envía un alerta administrativo a los administradores de la red. Al tercer alerta administrativo que se genere se le bloqueará la cuenta por un período de 15 días.
 - 9.3.5. ¿De qué manera nombra el sistema a cada unidad de disco?
 - 9.3.6. ¿Cómo identifica Ud. a qué unidad se hace referencia?
 - 9.3.7. ¿Podría Ud. indicar en que unidad y partición se encuentra instalado el GNU/Linux en su computadora? ¿Qué comandos o archivos de información utilizó?

10. VARIABLES DE ENTORNO

- 10.1. ¿Qué son las variables de entorno y para qué sirven?.
 - 10.1.1. Escriba el contenido y explique el significado de las siguientes variables: HOME / LOGNAME / PATH / HOSTNAME / IFS
 - 10.1.2. ¿Qué comando usó para ver el contenido de las variables del punto anterior?
 - 10.1.3. Cree una variable de entorno HOLA que contenga el mensaje "Hola mundo".
 - 10.1.4. ¿Cuál es el uso que le da el sistema a la variable PATH? ¿Qué ocurre si intenta ejecutar un comando que no se encuentra ubicado en alguno de los directorios que contiene la variable? ¿Cómo lo soluciona?
 - 10.1.5. ¿Por qué existen las variables PS1 y PS2? ¿Qué es un comando multilínea?

11. PLACA DE RED (En caso de no tener en su máquina, realizarlo en el Lab266)

- 11.1. ¿Para qué sirve el comando ifconfig y en que directorio se encuentra?
- 11.2. ¿Qué IP o IP's tiene asignada la computadora?
- 11.3. ¿Qué es el adaptador lo y para que se utiliza?
- 11.4. ¿Cuál es la salida del comando ping -c4 (ip del eth0)?

Trabajo Práctico Nro. 1 (GRUPAL):

- **Tema:** Programación de scripts básicos en PowerShell
 - **Descripción:** Se programarán todos los scripts mencionados en el presente trabajo, teniendo especialmente en cuenta las recomendaciones sobre programación mencionadas en la introducción de este trabajo.
 - **Formato de entrega:** Siguiendo el protocolo especificado anteriormente
 - **Documentación:** Todos los scripts que se entreguen deben tener un encabezado y un fin de archivo. Dentro del encabezado deben figurar el nombre del script, el trabajo práctico al que pertenece y el número de ejercicio dentro del trabajo práctico al que corresponde, el nombre de cada uno de los integrantes detallando nombre y apellido y el número de DNI de cada uno (tenga en cuenta que para pasar la nota final del trabajo práctico será usada dicha información, y no se le asignará la nota a ningún alumno que no figure en todos los archivos con todos sus datos), también deberá indicar el número de entrega a la que corresponde (entrega, primera reentrega, segunda reentrega, etc.). Para su mejor comprensión del tema vea el ejercicio 4 de la primera entrega que fue diseñado para que su trabajo sea más fácil.
 - **Evaluación:** Luego de entregado el trabajo práctico los ayudantes procederán a evaluar los ejercicios resueltos, en caso de encontrar errores se documentará en la carátula del TP que será devuelta al grupo con la evaluación final del TP y una fecha de reentrega en caso de ser necesaria (en caso de no cumplir con dicha fecha de reentrega el trabajo práctico será desaprobado). Cada ayudante podrá determinar si un determinado grupo debe o no rendir coloquio sobre el trabajo práctico presentado.
Las notas sobre los trabajos también estarán disponibles en el sitio de la cátedra (www.sisop.com.ar) donde además del estado de la corrección se podrá ver un detalle de las pruebas realizadas y los defectos encontrados
- Importante:** Cada ejercicio cuenta con una lista de validaciones mínimas que se realizará, esto no implica que se puedan hacer otras validaciones al momento de evaluar el trabajo presentado
- **Fecha de entrega:** Del **05/09/2016 al 09/09/2016**, dependiendo del día que se curse la materia y sólo en la primer fecha en que se cursa (lunes / martes / viernes).
 - **Fecha tardía de entrega:** Del **12/09/2016 al 16/09/2016**, dependiendo del día que se curse la materia y sólo en la primer fecha en que se cursa (lunes / martes).

Introducción:

La finalidad del presente práctico es que los alumnos adquieran un cierto entrenamiento sobre la programación de shell scripts ya sean en lenguaje de bash o PowerShell, practicando el uso de utilitarios comunes provistos por las dos grandes familias de sistemas operativos (GNU/Linux y Windows). Todos los scripts, están orientados a la administración de una máquina, o red y pueden ser interrelacionados de tal manera de darles una funcionalidad real.

Cabe destacar que todos los scripts deben poder ser ejecutados en forma batch o interactiva, y que en ningún caso serán probados con el usuario root o administrator, por lo cual deben tener en cuenta al realizarlos, no intentar utilizar comandos, directorios, u otros recursos que solo están disponibles para dicho usuario, o usuarios del grupo. Todos los scripts deberán funcionar en las instalaciones del laboratorio 266, ya que es ahí donde serán controlados por el grupo docente.

Para un correcto y uniforme funcionamiento, todos ellos deberán respetar algunos lineamientos generales:

1. Modularidad:

Si bien es algo subjetivo del programador, se trata de privilegiar la utilización de funciones (internas / externas), para lograr una integración posterior menos trabajosa.

2. Claridad:

Se recomienda fuertemente el uso de comentarios que permitan la máxima legibilidad posible de los scripts.

3. Verificación:

Todos los scripts deben realizar un control de las opciones que se le indiquen por línea de comando, es decir, verificar la sintaxis de la misma. En caso de error u omisión de opciones, al estilo de la mayoría de los comandos deben indicar mensajes como:

```
"Error en llamada!"  
Uso: comando [...]"
```

Donde se indicará entre corchetes [], los parámetros opcionales; y sin ellos los obligatorios, dando una breve explicación de cada uno de ellos.

Todos los scripts deben incluir una opción standard '-?' que indique el número de versión y las formas de llamada.

Ejercicios:

Los ejercicios que se encuentran dentro de esta sección fueron diseñados para que los alumnos adquieran un primer contacto con los comandos de PowerShell y con la forma de programar scripts con herramientas específicas del sistema operativo.

Ejercicio 1:

Dado el siguiente script:

```
Param($pathsalida)  
  
$existe = Test-Path $pathsalida  
if ($existe -eq $true)  
{  
    $lista = Get-ChildItem -File  
  
    foreach ($item in $lista)  
    {  
        Write-Host "$($item.Name)    $($item.Length)"  
    }  
}  
else  
{  
    Write-Error "El path no existe"  
}
```

Responda:

- ¿Cuál es el objetivo del script?
- ¿Qué validaciones agregaría a la definición de parámetros?
- ¿Con qué cmdlet se podría reemplazar el script para mostrar una salida similar?

Ejercicio 2:

Realizar un script que muestre el porcentaje de ocurrencia de cada carácter en un archivo cuya ruta será pasada por parámetro. Por ejemplo para un archivo con el siguiente contenido:

12bbb3hhhB

El resultado debería ser:

| | |
|---|-----|
| 1 | 10% |
| 2 | 10% |
| 3 | 10% |
| B | 10% |
| b | 30% |
| h | 30% |

Consideraciones:

- Los resultados deben mostrarse en formato de tabla utilizando el cmdlet Format-Table.
- Se debe distinguir entre mayúsculas y minúsculas.
- Se deben tener en cuenta todos los caracteres del archivo, incluidos los espacios, tabs, salto de línea, etc.

Criterios de corrección:

| Control | Criticidad |
|--|-------------|
| Debe cumplir con el enunciado | Obligatorio |
| El script debe tener ayuda visible con Get-Help | Obligatorio |
| Validación correcta de parámetros | Obligatorio |
| Uso de hash-tables (arrays asociativos) y Format-Table | Obligatorio |

Ejercicio 3:

Para poder realizar una búsqueda más rápida dentro de un directorio con miles de archivos, a un administrador de sistemas se le ocurrió ordenar los archivos dentro de subdirectorios cuyos nombres se encuentran basados en las primeras letras de los archivos, formando de esta manera un índice.

Se requiere realizar un script que ordene archivos siguiendo las siguientes reglas:

- Si la longitud del nombre del archivo es menor o igual a X caracteres (sin contar la extensión), el archivo no se debe mover.
- Si la longitud del nombre del archivo es mayor a X caracteres (sin contar la extensión), se debe mover dicho archivo a un subdirectorio cuyo nombre será la cadena formada por los primeros X caracteres del nombre del archivo. En el caso de no existir, crear el subdirectorio.
- Si los primeros X caracteres del nombre de un archivo son espacios, el archivo no se debe mover.

El script debe recibir por parámetro la ruta del directorio que contiene los archivos a ordenar y la cantidad X de caracteres.

Criterios de corrección:

| Control | Criticidad |
|--|-------------|
| Debe cumplir con el enunciado | Obligatorio |
| El script debe tener ayuda visible con Get-Help | Obligatorio |
| Validación correcta de parámetros | Obligatorio |
| Los archivos pueden tener cualquier nombre permitido por Windows, esto incluye nombres de archivo que contengan "[", "(", "\$", etc. | Obligatorio |
| Los subdirectorios deben crearse en la misma ruta donde se encuentran los archivos | Obligatorio |
| Junto con el script se deberá entregar un zip conteniendo la estructura de directorios y archivos utilizados para probar (antes de ser ejecutado el proceso) | Obligatoria |

Ejercicio 4:

Realizar un script que permita filtrar, ordenar y opcionalmente mostrar por pantalla una colección de objetos. Puede recibir los siguientes parámetros:

- **-propiedad:** Nombre de la propiedad sobre la que se va a operar. Obligatorio.
- **-filtro:** Valor que se debe buscar en la propiedad indicada con "-propiedad". El valor puede ser exacto o estar contenido en la misma. Obligatorio.
- **-desc:** Ordena de forma descendente. Opcional.
- **-asc:** Ordena de forma ascendente. Opcional.
- **-print:** Imprime la lista de propiedades que se pasan luego del parámetro (ej. -print ProcessName, Id). En caso de no ser suministrado este parámetro se deberá retornar la colección de objetos ordenada y filtrada para poder ser utilizada por otro script o cmdlet.

Criterios de corrección:

| Control | Criticidad |
|--|-------------|
| Debe cumplir con el enunciado | Obligatorio |
| El script debe tener ayuda visible con Get-Help | Obligatorio |
| Validación correcta de parámetros | Obligatorio |
| Los parámetros -desc y -asc no pueden ser usados en simultáneo | Obligatorio |
| Se debe usar Where-Object, Select-Object y Sort-Object | Obligatorio |
| La entrada del script debe ser a través del pipeline (utilizado " ") | Obligatorio |

Ejercicio 5:

Realizar un script que escriba en un archivo, cada N segundos, un listado de los M procesos que más utilización de memoria tienen, especificando por cada uno de ellos la siguiente información:

Identificador (PID) – Path del ejecutable – Memoria (Working Set).

Tanto N como M deben ser pasados como parámetros al script, si N es igual a 0, entonces la información deberá guardarse sólo una vez. En caso de N mayor a cero, la información se actualizará cada N segundos.

El intervalo de N segundos se debe controlar con un objeto Timer (System.Timers.Timer). No se puede utilizar sleep.

Criterios de corrección:

| Control | Criticidad |
|--|-------------|
| Debe cumplir con el enunciado | Obligatorio |
| El script debe tener ayuda visible con Get-Help | Obligatorio |
| Validación correcta de parámetros | Obligatorio |
| Se debe usar el cmdlet Get-WmiObject | Obligatorio |
| Se debe usar un timer (System.Timers.Timer) para contar el tiempo entre cada una de las escrituras en el archivo | Obligatorio |
| El script debe quedar corriendo en segundo plano hasta que se elimine el evento del timer | Obligatorio |
| No se utiliza Sleep | Obligatorio |
| No debe realizar esperas activas | Obligatorio |

Ejercicio 6:

Realizar un script que a partir de un archivo .zip pasado por parámetro indique la relación de compresión de cada uno de los archivos que contiene. Ejemplo de salida:

| Nombre archivo | Tamaño original | Tamaño comprimido | Relación |
|----------------|-----------------|-------------------|----------|
| Archivo1.txt | 100 | 10 | 0,1 |
| Archivo2.jpg | 2366 | 2254 | 0,95 |

Criterios de corrección:

| Control | Criticidad |
|--|-------------|
| Debe cumplir con el enunciado | Obligatorio |
| El script debe tener ayuda visible con Get-Help | Obligatorio |
| Validación correcta de parámetros | Obligatorio |
| Usar Add-Type | Obligatorio |
| Utilizar la clase System.IO.Compression.ZipFile para resolver el ejercicio | Obligatorio |
| Los tamaños deben estar expresados en MB | Obligatorio |
| Las relaciones deben tener tres decimales máximo | Obligatorio |

Ejercicio 7 (Entrega tardía):

Realizar un script que realice un log de todos los procesos que inician y terminan. El log debe informar la fecha del evento, el tipo de evento (inicio o finalización) y el nombre del proceso. El nombre del archivo de log se debe pasar por parámetro.

Criterios de corrección:

| Control | Criticidad |
|---|-------------------|
| Debe cumplir con el enunciado | Obligatorio |
| El script debe tener ayuda visible con Get-Help | Obligatorio |
| Validación correcta de parámetros | Obligatorio |
| Se deben usar eventos para realizar el script | Obligatorio |
| No debe realizar esperas activas | Obligatorio |

Trabajo Práctico Nro. 2 (GRUPAL):

- **Tema:** Programación de scripts en tecnología bash
- **Descripción:** Se programarán todos los scripts mencionados en el presente trabajo, teniendo especialmente en cuenta las recomendaciones sobre programación mencionadas en la introducción de este trabajo.
- **Formato de entrega:** Siguiendo el protocolo especificado anteriormente
- **Documentación:** Todos los scripts que se entreguen deben tener un encabezado y un fin de archivo. Dentro del encabezado deben figurar el nombre del script, el trabajo práctico al que pertenece y el número de ejercicio dentro del trabajo práctico al que corresponde, el nombre de cada uno de los integrantes detallando nombre y apellido y el número de DNI de cada uno (tenga en cuenta que para pasar la nota final del trabajo práctico será usada dicha información, y no se le asignará la nota a ningún alumno que no figure en todos los archivos con todos sus datos), también deberá indicar el número de entrega a la que corresponde (entrega, primera reentrega, segunda reentrega, etc.).
- **Evaluación:** Luego de entregado el trabajo práctico los ayudantes procederán a evaluar los ejercicios resueltos, en caso de encontrar errores se documentará en la carátula del TP que será devuelta al grupo con la evaluación final del TP y una fecha de reentrega en caso de ser necesaria (en caso de no cumplir con dicha fecha de reentrega el trabajo práctico será desaprobado). Cada ayudante podrá determinar si un determinado grupo debe o no rendir coloquio sobre el trabajo práctico presentado.

Las notas sobre los trabajos también estarán disponibles en el sitio de la cátedra (www.sisop.com.ar) donde además del estado de la corrección se podrá ver un detalle de las pruebas realizadas y los defectos encontrados

Importante: Cada ejercicio cuenta con una lista de validaciones mínimas que se realizará, esto no implica que se puedan hacer otras validaciones al momento de evaluar el trabajo presentado

- **Fecha de entrega:** Del **26/09/2016 al 30/09/2016**, dependiendo del día que se curse la materia y sólo en la primer fecha en que se cursa (lunes / martes / viernes).
- **Fecha tardía de entrega:** Del **03/10/2016 al 07/10/2016**, dependiendo del día que se curse la materia y sólo en la primer fecha en que se cursa (lunes / martes).
- **Introducción:**

La finalidad del presente práctico es que los alumnos adquieran un cierto entrenamiento sobre la programación de shell scripts, practicando el uso de utilitarios comunes provistos por los sistemas operativos (GNU/Linux). Todos los scripts, están orientados a la administración de una máquina, o red y pueden ser interrelacionados de tal manera de darles una funcionalidad real.

Cabe destacar que todos los scripts deben poder ser ejecutados en forma batch o interactiva, y que en ningún caso serán probados con el usuario root, por lo cual deben tener en cuenta al realizarlos, no intentar utilizar comandos, directorios, u otros recursos que solo están disponibles para dicho usuario. Todos los scripts deberán funcionar en las instalaciones del laboratorio 266, ya que es ahí donde serán controlados por el grupo docente.

Para un correcto y uniforme funcionamiento, todos ellos deberán respetar algunos lineamientos generales indicados en el trabajo práctico anterior

Ejercicios:

Tip: En caso de tener problemas con un script bajado desde un dispositivo formateado con DOS, y que contiene ^M al final de cada línea puede usar el siguiente comando para eliminarlos:

```
tr -d '\r' <archivo_con_M >archivo_sin_M
```

En caso de ejecutar scripts que usen el archivo de passwords, en el laboratorio, debe cambiar "cat /etc/passwd" por "getent passwd"

Ejercicio 1:

Tomando en cuenta las siguientes ejecuciones de scripts responda las preguntas que se encuentran más abajo. Tenga en cuenta que antes de poder ejecutarlos deberá marcar ambos scripts como ejecutables.

Importante: como parte del resultado se deberá entregar los script en archivos tipo sh y las respuestas en el script2.sh como comentarios.

```
$ cat script1.sh
#!/bin/bash
variable=$1
echo "valor $variable"

$ cat script2.sh
#!/bin/bash
./script1.sh 10
echo "Resultado $variable"
. script1.sh 15
echo "Resultado $variable"
./script1.sh $$
echo 'Resultado $variable'
script1.sh 35
echo "Resultado $variable"
```

NOTA: Todos los scripts están probados y deberán ser copiados textualmente respetando espacios y signos de puntuación tal como son presentados.

Responda:

- ¿Qué tuvo que realizar para poder ejecutar los scripts luego de generar los archivos? ¿con qué comando/s lo realizó?
- ¿Se presentó algún error al momento de ejecutar el script? ¿a qué se debe el error? ¿cómo se solucionaría? (indique al menos dos formas)
- Mencione como se pasan/emplean parámetros en scripts de bash.
- ¿Qué particularidad detecta dentro de script2 con la variable?
- ¿Qué significa la línea "#!/bin/bash"? ¿Cuáles son las diferentes formas de ejecutar un script?
- ¿Qué información brinda la variable "\$\$"? ¿Qué otras variables similares conoce? Explíquelas.
- Explique las diferencias entre los distintos tipos de comillas que se pueden utilizar en Shell scripts. Realice un ejemplo para tipo de comilla.

Ejercicio 2:

Crear un script para el control de la conexión de los usuarios.

Parámetros permitidos:

- u usuario: muestra las conexiones activas de un usuario en particular, si no se indica el usuario se debe tomar al usuario que ejecuto el script
- a: muestra los usuarios con conexiones activas y la última conexión de los usuarios sin conexiones activas
- c: muestra los usuarios y la cantidad de conexiones que tuvo cada uno.

Si no se indican parámetros trabajar como si se hubiese indicado -a.

La salida del script debe contener el siguiente formato:

| Usuario | Ult. Conexión | Tiempo de Conexión |
|----------|------------------|--------------------|
| Usuario1 | Activo en pts/0 | --- |
| Usuario2 | Activo en pts/1 | --- |
| Usuario2 | Activo en tty1 | --- |
| Usuario3 | 23/03/2015 23:50 | 00:20 hs |
| Usuario4 | 21/03/2015 18:35 | 03:42 hs |

Criterios de corrección:

| Control | Criticidad |
|---|-------------|
| Funciona correctamente según enunciado | Obligatorio |
| Validación de cantidad y tipo de parámetros. | Obligatorio |
| Incluir la opción -? o -h con la ayuda del modo de uso del script de forma clara y legible. | Obligatorio |
| Mostrar los mensajes de error correspondientes a cada validación | Obligatorio |
| Comentarios en el código | Obligatorio |
| Uso de funciones. | Opcional |

Ejercicio 3:

Realizar un script que genere un reporte de la cantidad de archivos ejecutables de un directorio para el usuario conectado (no se debe actuar sobre los links).

Podrá recibir 4 parámetros, no obligatorios:

- -d directorio: es el directorio a analizar, en caso de no enviarlo utilizará la variable PATH.
- -s archivo: es el archivo de salida de la información, con el siguiente formato: ruta/archivo (Tenga en cuenta que la ruta puede ser absoluta o relativa).
- -y: deberá informar por cada directorio, los subtotales de cantidad de ejecutables por año.
- -r: debe analizar también los subdirectorios, si existieran. En caso de no recibirlo solo analizará el directorio especificado sin subdirectorios.

Ejemplos de salida:

Salida para la variable PATH:

Usuario: cagarcia – Directorios analizados de PATH.

Cantidad total de comandos disponibles: 1500

Detalle de comandos disponibles por directorio:

```
/home/cagarcia/bin: 3
/usr/bin:          1403
/bin:              94
```

Salida para la variable PATH (con parámetro -y):

Usuario: cagarcia – Directorios analizados de PATH.

Cantidad total de comandos disponibles: 1500

Detalle de comandos disponibles por directorio:

```
/home/cagarcia/bin: 2014: 3
                   2013: 900
                   2014: 183
                   2015: 320
/bin:              2012: 54
                   2015: 40
```

Salida para directorio /HOME/.../usuario:

Usuario: cagarcia – Directorio analizado: /HOME/.../usuario.

Cantidad total de comandos disponibles: 150

Salida para directorio /HOME/.../usuario (con parámetro -r):

Usuario: cagarcia – Directorio analizado: /HOME/.../usuario.

Cantidad total de comandos disponibles: 150

Detalle de comandos disponibles por directorio:

```
... /desktop:      5
... /download:     95
... /documents:    50
```

Criterios de corrección:

| Control | Criticidad |
|---|-------------|
| Funciona correctamente según enunciado | Obligatorio |
| Validación de cantidad y tipo de parámetros. | Obligatorio |
| Incluir la opción -? o -h con la ayuda del modo de uso del script de forma clara y legible. | Obligatorio |
| No está permitido el uso de awk, ni de archivos temporales | Obligatorio |
| El reporte debe poseer el formato de salida ejemplificado | Obligatorio |
| Mostrar los mensajes de error correspondientes a cada validación | Obligatorio |
| Comentarios en el código | Obligatorio |
| Uso de funciones. | Opcional |

Ejercicio 4:

El sistema de facturación de una PyME guarda los registros de sus ventas en archivos diarios.

El nombre de los archivos mantiene el siguiente formato: ventas-dd.mm.aaaa (por ejemplo: ventas-29.02.2016)

Cada registro posee datos de la factura realizada y del cliente, organizados en la estructura que se muestra a continuación:

Hora|Codigo De Factura|Codigo De Cliente|Razon Social|Importe
(ej.: 13:00:00|543|3|Jose Perez|145,60)

Se requiere un script que obtenga un reporte de venta total del mes (opción -m). Este reporte puede recibir como valor, una fecha en formato mm/aaaa. De no recibirlo, se realizará el reporte del mes en curso.

Así mismo, debe poder obtenerse un reporte de venta detallada de un cliente (opción -c) para el mes actual. Se recibirá como valor el código del cliente y se deberá obtener un reporte que especifique cada compra realizada por el cliente, mostrando todos los datos que existan de la operación (incluyendo la fecha)

Criterios de corrección:

| Control | Criticidad |
|---|-------------|
| Funciona correctamente según enunciado | Obligatorio |
| Se debe validar el archivo de entrada | Obligatorio |
| Junto al script deberá entregar los archivos de ejemplo utilizados | Obligatorio |
| Los reportes deben poseer un formato de salida claro y legible | Obligatorio |
| Incluir la opción -? o -h con la ayuda del modo de uso del script de forma clara y legible. | Obligatorio |
| Comentarios en el código | Obligatorio |
| Uso de funciones. | Opcional |

Ejercicio 5:

Realizar un script para administrar archivos de configuración de un sistema operativo de la familia Unix.

El script debe recibir como mínimo tres parámetros: el archivo de configuración a modificar, la clave y por último el nuevo valor.

Ejemplo de uso:

Configurar.sh /etc/archivo.conf INIT "Nuevo Valor"

Si no existe la clave en el archivo de configuración se la deberá agregar al final del mismo, caso contrario se deberá mostrar el valor actual y solicitar la confirmación del cambio. Esta confirmación no será requerida en caso de que se haya recibido el parámetro -y

Todos los cambios deberán ser documentados en el mismo archivo de configuración, es decir que cuando se dé de alta el parámetro se deberá documentar el usuario y la fecha y hora en

la que se agregó, en cambio si se realiza la modificación se deberá documentar, además de lo anterior, el valor previo al cambio. Los comentarios con esta documentación se deberán encontrar inmediatamente antes de la clave con el valor.

Si se le envía el parámetro -c, deberá permitir al usuario incluir un comentario adicional, que será agregado junto a la documentación del agregado/cambio de la clave.

Ejemplo de uso:

```
Configurar.sh /etc/archivo.conf INIT "Nuevo Valor" -c "Comentario mas largo"
```

Importante: Tenga en cuenta que no se podrá trabajar en forma directa sobre los archivos de configuración del sistema (ya que no tiene los permisos necesarios), por lo que para realizar la prueba deberá copiarse los archivos que se quiera modificar a su home directory.

Criterios de corrección:

| Control | Criticidad |
|---|-------------|
| Funciona correctamente según enunciado | Obligatorio |
| Validación de cantidad y tipo de parámetros. | Obligatorio |
| Incluir la opción -? o -h con la ayuda del modo de uso del script de forma clara y legible. | Obligatorio |
| Mostrar los mensajes de error correspondientes a cada validación | Obligatorio |
| El archivo de configuración puede ser pasado por ruta absoluta o relativa al directorio de ejecución del script | Obligatorio |
| Uso del comando sed para resolver la problemática del ejercicio. | Obligatorio |
| Comentarios en el código | Obligatorio |
| Junto al script deberá entregar un archivo de configuración como prueba | Obligatorio |
| Uso de funciones. | Opcional |

Ejercicio 6:

Realizar un script que ejecute como proceso demonio para controlar el tamaño de los archivos generados en un directorio y subdirectorios del mismo. Este proceso deberá finalizar cuando se le envíe una señal SIGUSR1 y no deberá finalizar con SIGINT.

Inicialmente debe indicar el tamaño que posee cada tipo de archivo (determinado por la extensión) que se encuentran en el árbol del directorio.

Cada 30 segundos debe controlar si hubo un cambio en el directorio, en caso afirmativo indicar el nuevo tamaño y el porcentaje de incremento/decremento.

La salida debe actualizarse automáticamente cada vez que se encuentre un cambio. Así como también si se generan archivos nuevos con extensiones no listadas con anterioridad o todos los archivos de una extensión eliminados en su totalidad.

La ruta del archivo de salida y el archivo de salida deben poder ser parametrizables (parámetro 1 del script), con el siguiente formato: ruta/archivo (Tenga en cuenta que la ruta puede ser absoluta o relativa). De no especificarse parámetro, el script mostrará el reporte por pantalla

Formato de la salida esperada: (PID del demonio, fecha y hora, extensión, tamaño, observaciones)

```
1525 01/03/2015 08:01:00 jpg 100mb
1525 01/03/2015 08:01:00 png 200mb
1525 01/03/2015 08:09:30 jpg 162mb (+62%)
1525 01/03/2015 08:09:30 png 100mb (-50%)
1525 01/03/2015 08:15:00 avi 250mb (Nueva)
1525 01/03/2015 08:15:00 jpg 0mb (Eliminada)
```

Criterios de corrección:

| Control | Criticidad |
|--|-------------|
| Funciona correctamente según enunciado | Obligatorio |
| Validación de cantidad y tipo de parámetros. | Obligatorio |
| Incluir la opción -? o -h con la ayuda del modo de uso del script de forma clara y legible. | Obligatorio |
| Mostrar los mensajes de error correspondientes a cada validación | Obligatorio |
| El control se debe realizar empleando un demonio. No puede haber más de un demonio controlando el mismo árbol de directorio | Obligatorio |
| Se pueden controlar dos directorios distintos en forma concurrente | Obligatorio |
| Maneja correctamente las señales | Obligatorio |
| Comentarios en el código | Obligatorio |
| Elimina correctamente todos los archivos temporales utilizados | Obligatorio |
| Uso de funciones. | Opcional |

Ejercicio 7 (Entrega tardía):

Realizar un script para generar resúmenes mensuales de tarjetas de crédito a partir de archivos de gastos detallados.

El script deberá analizar el directorio actual en busca de archivos de gastos mensuales. Éstos tendrán como nombre el mes y año con extensión .in (por ejemplo para marzo de 2015, será 201503.in).

Dichos archivos estarán compuestos por registros con el siguiente formato:

Fecha(dd/mm/aaaa), descripción, cantidad de cuotas, importe total.

Ejemplo para el archivo **201503.in**:

12/03/2015,ypf Premium, 1,530

13/03/2015,wrangleer jeans,3,750

25/03/2015,electrodomésticos FR,2,660

Los resúmenes generados deberán quedar almacenados en archivos con extensión .res y nombrados con el año y mes (por ejemplo para marzo de 2015, será: 201503.res), y el formato de los registros será como el siguiente:

Fecha(dd/mm/aaaa), descripción, cuota x/total de cuotas, importe de la cuota.

Para generar esta información se deben considerar todos los gastos de cada mes que se encuentren en el directorio.

Importante: Esto último incluye los gastos en cuotas. Ejemplo: Si consideramos el archivo 201503.in, deberemos generar el resumen de marzo, con la primera cuota de cada gasto y además generar el resumen de abril, que incluirá la segunda cuota de los gastos correspondientes y el de mayo con la cuota restante.

Como último registro de estos archivos .res se deberá incluir un total.

Se muestra un ejemplo completo:

Ejemplo para el archivo 201503.in, se generarán los siguientes archivos:

Archivo **201503.res**, cuyo contenido será:

12/03/2015,ypf Premium, 1/1, 530

13/03/2015,pantalones jines, 1/3, 250

25/03/2015,electrodomésticos FR, 1/2, 330

28/03/2015,Total a pagar,,1110

Archivo **201504.res**, cuyo contenido será:

13/03/2015,pantalones jines, 2/3, 250

25/03/2015,electrodomésticos FR, 2/2, 330

28/04/2015,Total a pagar,,580

Archivo **201505.res**, cuyo contenido será:

13/03/2015,pantalones jines, 3/3, 250

28/05/2015,Total a pagar,,250

Los parámetros podrá recibir el script (uno a la vez) serán:

-a: realizar el análisis y generación de los resúmenes, deberá validar que no existan resúmenes previamente, sino informarlo y detener la ejecución.

-d: borrará del directorio actual todos los resúmenes existentes (archivos .res). Deberá pedir confirmación de borrado ("¿Desea borrar los archivos? Si / No)

-m [mmaaaa]: mostrará por pantalla, con formato estilo tabla, el detalle del resumen mensual del mes pasado por parámetro (ej: 042015), en caso de no pasar nada, será del mes y año actuales.

Nota: en la carpeta de ejecución, puede haber uno o varios archivos de entrada (gastos mensuales) pero nunca habrá dos del mismo mes/año. Si algún mes no tiene resumen, ya que no hay gastos, no se deberá generar el archivo de dicho mes.

En la carpeta del script, al finalizar la ejecución, solo deberán estar los archivos de entrada, los generados y el script. No deberá haber archivos temporales.

Criterios de corrección:

| Control | Criticidad |
|--|-------------|
| Funciona correctamente según enunciado | Obligatorio |
| Validación de cantidad y tipo de parámetros. | Obligatorio |
| Incluir la opción -? con la ayuda del modo de uso del script. | Obligatorio |
| Mostrar los mensajes de error correspondientes a cada validación | Obligatorio |
| Controla e informa si ya existen resúmenes antes de ejecutar con parámetro -a. | Obligatorio |
| Controla los permisos sobre los archivos y directorio donde se almacena la información | Obligatorio |
| Entregar los archivos de ejemplo utilizados para validar el correcto funcionamiento | Obligatorio |
| Comentarios en el código | Obligatorio |
| Uso de funciones. | Opcional |
| Utiliza código awk y/o sed. | Deseable |

Trabajo Práctico Nro. 3 (GRUPAL):

- **Tema:** Diagnósticos, virtualización y storage
- **Descripción:** El presente trabajo práctico tiene como objetivo realizar el análisis del comportamiento de algunos procesos a desarrollar por el grupo para comprender y evaluar el comportamiento de los procesos y como se ven afectados por la configuración del sistema operativo.

Es importante mencionar que todas las evaluaciones deberán realizarse sobre el mismo hardware para que las pruebas sean más significativas y den resultados mucho más realistas.

Es importante mencionar que si bien se debe codificar en C, la forma de codificar no será parte de la evaluación del TP, se solicita que sea entregada pero no formará parte de la nota.
- **Formato de entrega:** Siguiendo el protocolo especificado anteriormente
- **Evaluación:** Luego de entregado el trabajo práctico los ayudantes procederán a evaluar resultados obtenidos por los grupos ante cada uno de los ejercicios planteados y procederá a evaluar individual o grupalmente a cada uno de los integrantes o grupos validando sus presentaciones. Este trabajo práctico cuenta con coloquio obligatorio y para todos los integrantes del grupo, por lo que el grupo docente evaluador coordinará una fecha con cada uno de los grupos para realizar la evaluación correspondiente (pudiendo esta ser oral o escrita).

Las notas sobre los trabajos también estarán disponibles en el sitio de la cátedra (www.sisop.com.ar) donde además del estado de la corrección se podrá ver un detalle de las pruebas realizadas y los defectos encontrados.

IMPORTANTE: Solo se evaluarán los trabajos prácticos que estén entregados en forma completa (incluyendo todos los puntos solicitados), caso contrario el trabajo práctico no será considerado como entregado.
- **Fecha de entrega:** Del **07/11/2016 al 11/11/2016**, dependiendo del día que se curse la materia y sólo en la primer fecha en que se cursa (lunes / martes).

Ejercicios:

Todos los ejercicios deben ser codificados en C o C++ bajo sistema operativo Linux. Todos los programas realizados deberán ser entregados como parte de la documentación del trabajo práctico pero no serán evaluados más que en su funcionalidad.

No es necesaria la generación de herramientas de compilación (Makefiles), ni el uso de bibliotecas privadas para compartir

Ejercicio 1: Performance de procesos pesados y livianos

Implemente el siguiente algoritmo y responda las preguntas sobre performance:

- ✓ Un proceso que reciba por parámetro la cantidad de posiciones de un array y actúe en base a esta cantidad. Se deberá ejecutar cada proceso al menos tres veces con los valores 25000, 50000, 75000
- ✓ Crear un array de enteros de X (el parámetro) posiciones e inicializarlo con números random
- ✓ Tomar estructura de estadísticas del proceso iniciales
 - Usar `syscall getrusage()`
- ✓ Tomar tiempo reloj de inicio
 - Usar `clock_gettime()` con el timer `CLOCK_MONOTONIC_RAW`
- ✓ Generar un número random único, que será utilizado posteriormente por las unidades de procesamiento, que no sea ni 0 y 1
- ✓ Lanzar 1000 unidades de procesamiento de a una por vez (el proceso principal debe crear la unidad de procesamiento y esperar a que finalice su trabajo)
- ✓ El objetivo de cada unidad de procesamiento será operar sobre el array definido e inicializado por el proceso principal
- ✓ Se deberán generar los siguientes casos:
 - Caso 1:
Unidad de procesamiento: Proceso pesado
Acceso a datos: Lectura
 - Caso 2:
Unidad de procesamiento: Proceso pesado

- Acceso a datos: Escritura
- Caso 3:
Unidad de procesamiento: Proceso liviano
Acceso a datos: Lectura
- Caso 4:
Unidad de procesamiento: Proceso liviano
Acceso a datos: Escritura
- ✓ Tomar estructura de estadísticas del proceso hijo
 - Usar el syscall `wait3()` o `wait34()` que devuelve la estructura `rusage` con los datos estadísticos del proceso
- ✓ Tomar tiempo reloj de fin
- ✓ Calcular las estadísticas finales, para poder sumarizar los parciales y obtener los valores promedios por cada ejecución
- ✓ Imprimir **todas** las estructuras de datos utilizadas para calcular las estadísticas (no el array)

El acceso a datos define el tipo de operación que se realizará en cada uno de los casos.:

- Lectura: Realiza la sumatoria de las X posiciones del array en una variable
- Escritura: Multiplica cada posición del array por el número random único generado inicialmente, y guardarlo en la misma posición

Aclaraciones generales:

- No se evaluará el código fuente ni la forma en el que se encuentra programado. Solo se analizará que esté cumpliendo con el objetivo funcional
- No se debe realizar ningún tipo de salida de los datos manipulados en el array. El objetivo es evaluar el uso de recursos y no las operaciones aritméticas realizadas
- Las estadísticas de procesos son calculadas por el sistema operativo para cada proceso pesado. Al ejecutar un proceso hijo se deberán obtener las estadísticas del proceso hijo y no las del padre que lo lanzó (ver syscall `wait3` y `wait4`)

Entregables:

- Código fuente utilizado para evaluar cada uno de los casos
- Estructura de datos que se imprime como resultados de cada procesamiento
- Completar la siguiente tabla:

| Métrica | Caso 1 | | | Caso 2 | | | Caso 3 | | | Caso 4 | | |
|--|--------|----|----|--------|----|----|--------|----|----|--------|----|----|
| | C1 | C2 | C3 | C1 | C2 | C3 | C1 | C2 | C3 | C1 | C2 | C3 |
| Tiempo reloj | | | | | | | | | | | | |
| Tiempo reloj promedio* *por unidad de procesamiento | | | | | | | | | | | | |
| Tiempo CPU sistema total | | | | | | | | | | | | |
| Tiempo CPU usuario total | | | | | | | | | | | | |
| Tiempo CPU sistema promedio | | | | | | | | | | | | |
| Tiempo CPU usuario promedio | | | | | | | | | | | | |
| Cantidad de Soft Page Faults | | | | | | | | | | | | |
| Cantidad de Hard Page Faults | | | | | | | | | | | | |
| Cantidad de señales recibidas | | | | | | | | | | | | |
| Cambios de contexto voluntarios | | | | | | | | | | | | |
| Cambios de contexto involuntarios | | | | | | | | | | | | |

➤ Conclusiones:

- ✓ Analice el comportamiento y los datos obtenidos en cada uno de los casos, compare los resultados entre los mismos y explique a qué se deben las diferencias y similitudes en los comportamientos observados
- ✓ Indique qué conceptos teóricos se ven reflejados

Criterios de corrección:

| Control | Criticidad |
|---|-------------|
| Cumple con lo solicitado en el enunciado | Obligatorio |
| Los resultados confirman los conceptos teóricos vistos durante la cursada | Obligatorio |
| Están respondidas las conclusiones | Obligatorio |
| Se entrega el código fuente de cada uno de los procesos | Obligatorio |
| Código legible y documentado | Deseable |

Ejercicio 2: Performance de datos compartidos

Implemente el siguiente algoritmo y responda las preguntas sobre performance:

- ✓ Crear dos procesos A (productor) y B (consumidor), no emparentados para que se comuniquen entre si
- ✓ Crear la estructura de comunicación basada en uno de los métodos a evaluar. La estructura se debe abrir en los dos procesos
- ✓ Tomar estructura de estadísticas del proceso iniciales
 - Usar `syscall getrusage()` que devuelve la estructura `rusage` con los datos estadísticos del proceso
- ✓ Tomar tiempo reloj de inicio
 - Usar `clock_gettime()` con el timer `CLOCK_MONOTONIC_RAW`
- ✓ Intercambiar 1000 mensajes (no importa el contenido) de 1 KB a través de la estructura de comunicación. Los mensajes deben ir y volver del proceso A y B y viceversa en una alternancia estricta (ping-pong)
- ✓ Se deberán generar los siguientes casos:
 - Caso 1:
Estructura de comunicación: Archivos
 - Caso 2:
Estructura de comunicación: FIFO (un solo recurso bidireccional)
 - Caso 3:
Estructura de comunicación: Memoria Compartida
 - Caso 4:
Estructura de comunicación: Sockets (solo tipo TCP)
Procesos: Los dos en la misma máquina utilizando localhost
 - Caso 5:
Estructura de comunicación: Sockets (solo tipo TCP)
Procesos: Los dos en la misma máquina utilizando IP local (no localhost)
 - Caso 6:
Estructura de comunicación: Sockets (solo tipo TCP)
Procesos: En máquinas separadas (conectadas por red)
- ✓ Tomar estructura de estadísticas del proceso final
- ✓ Tomar tiempo reloj de fin
- ✓ Calcular las estadísticas finales
- ✓ Imprimir **todas** las estructuras de datos utilizadas para calcular las estadísticas

Aclaraciones generales:

- No se evaluará el código fuente ni la forma en el que se encuentra programado. Solo se analizará que esté cumpliendo con el objetivo funcional
- No se debe realizar ningún tipo de salida de los mensajes intercambiados y no importa su contenido, solo que sean del tamaño indicado. El objetivo es evaluar el uso de recursos y no las operaciones realizadas
- Las mediciones solo deben ser tomadas solo en el proceso A
- Para los casos 1 y 2 el recurso de comunicación de los datos debe ser abierto al inicio y cerrado a la finalización y no en cada ciclo. Si se necesita un recurso de sincronización su utilización es libre
- Solo se pueden utilizar semáforos para el caso 3
- El código fuente y binario de los casos 4, 5 y 6 debe ser el mismo (la IP debe ser pasada por parámetro) solo debe diferir la forma de ejecución
- Para el caso 6 se pueden utilizar máquinas virtuales siempre y cuando se encuentren ejecutando en dos máquinas reales distintas (uso de networking obligatoria)

Entregables:

- Código fuente utilizado para evaluar cada uno de los casos
- Estructura de datos que se imprime como resultados de cada procesamiento
- Completar la siguiente tabla:

| Métrica | Caso 1 | Caso 2 | Caso 3 | Caso 4 | Caso 5 | Caso 6 |
|-------------------------------------|--------|--------|--------|--------|--------|--------|
| Tiempo reloj | | | | | | |
| Tiempo CPU sistema total | | | | | | |
| Tiempo CPU usuario total | | | | | | |
| Cantidad de Soft Page Faults | | | | | | |
| Cantidad de Hard Page Faults | | | | | | |
| Operaciones de Entrada (en bloques) | | | | | | |
| Operaciones de Salida (en bloques) | | | | | | |

➤ **Conclusiones:**

- ✓ Analice el comportamiento y los datos obtenidos en cada uno de los casos, compare los resultados entre los mismos y explique a qué se deben las diferencias y similitudes en los comportamientos observados
- ✓ Indique cuál/es métodos pueden ser utilizados en entornos distribuidos (más de una computadora). Indique qué características tiene que tener la instalación para que esto suceda
- ✓ Indique que conceptos teóricos están involucrados en cada uno de los casos

Criterios de corrección:

| Control | Criticidad |
|---|-------------|
| Cumple con lo solicitado en el enunciado | Obligatorio |
| Están respondidas las conclusiones | Obligatorio |
| Se entrega el código fuente de cada uno de los procesos | Obligatorio |
| Código legible y documentado | Deseable |

Ejercicio 3: Performance de tecnologías de almacenamiento

Implemente en un servidor virtual SAN & NAS, basado en FreeNAS, sobre un virtualizador de la familia VMWare los siguientes tipos de RAID y realice la comparativa de performance de cada uno de ellos.

Tecnologías de RAID a implementar (cada RAID debe estar conformado por 6 discos de al menos 200 MB con una única partición cada uno y configurados como Fixed Size en el virtualizador):

- RAID 0
- RAID 1
- RAID 5
- RAID 6
- RAID 10

Compartir los RAIDs y evaluar el sistema:

- Los RAIDs generados deberán ser compartidos a los sistemas operativos cliente a través de los siguientes protocolos:
 - Protocolo orientado a red (NAS).
 - Protocolo orientado a bloques (SAN).
- La evaluación del sistema deberá ser realizada en el sistema operativo cliente para todos los RAIDs y protocolos con un proceso (script, programa c, etc.) que escriba un archivo de aproximadamente 150 MB (el mismo proceso para todas las pruebas)

Entregables:

- Completar la siguiente tabla:

| Caso | Tiempo Reloj | Tiempo Usuario | Tiempo Sistema | Bloques Entrada | Bloques Salida |
|--------------|--------------|----------------|----------------|-----------------|----------------|
| RAID 0 – NAS | | | | | |

| | | | | | |
|---------------|--|--|--|--|--|
| RAID 0 – SAN | | | | | |
| RAID 1 – NAS | | | | | |
| RAID 1 – SAN | | | | | |
| RAID 5 – NAS | | | | | |
| RAID 5 – SAN | | | | | |
| RAID 6 – NAS | | | | | |
| RAID 6 – SAN | | | | | |
| RAID 10 – NAS | | | | | |
| RAID 10 – SAN | | | | | |

➤ **Conclusiones:**

- Analice el comportamiento y los datos obtenidos en cada uno de los casos, compare los resultados entre los mismos y explique a qué se deben las diferencias y similitudes en los comportamientos de cada uno de los RAID según la forma en los que fueron compartidos.
- A qué se debe que se haya solicitado que los discos virtuales se definan como Fixed Size
- **Tolerancia a fallos:** demuestre a través de imágenes (capturas de pantallas) del producto SAN & NAS el punto de quiebre que cada uno de los RAIDs pudieron soportar desde que el RAID se encuentra en estado “Clean” hasta la pérdida total del recurso (si soporta la rotura de 5 discos se deberán incluir las 5 imágenes eliminando cada uno de los discos).
Todas las imágenes deben ser incluidas en un documento tipo Word con una breve explicación sobre lo que se está demostrando. No recortar las imágenes ya que se debe visualizar toda la información suministrada por el producto SAN & NAS en su página de administración incluida la url.
Importante: no se aceptarán videos como documentación.
- Indique qué conceptos teóricos están involucrados en cada uno de los casos anteriores.

Criterios de corrección:

| Control | Criticidad |
|--|-------------|
| Cumple con lo solicitado en el enunciado | Obligatorio |
| Están respondidas las conclusiones | Obligatorio |
| Se entrega el código fuente del proceso solicitado | Obligatorio |
| Las imágenes de tolerancia a fallos son completas y muestran la evolución completa desde el recurso en estado limpio hasta la rotura total | Obligatorio |
| Código legible y documentado | Deseable |