



PROYECTO N° 1

SISTEMAS OPERATIVOS

Licenciatura en Ciencias de la Computación - Segundo cuatrimestre de 2010
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



1) Como compilar utilizando gcc y make

- a) Investigue como se confeccionan los archivos *Makefile* para poder compilar con el comando: `make` (el cual deberá utilizar en cada uno de los programas en lenguaje C en este proyecto sobre el sistema operativo Linux).

2) Sobre sistema operativo Windows NT

- a) Explique y ejemplifique cómo se crea un proceso. Muestre las relaciones que hay con la creación de proceso en sistemas operativos de la familia UNIX.
- b) ¿Cómo puede visualizar la jerarquía de procesos en ejecución en un determinado momento en el sistema?
- c) Instale y experimente con los siguientes programas:
 - Sysinternals Process Monitor:
<http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>
 - Sysinternals Process Explorer:
<http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx>

Explique en no más de 10 líneas (para cada uno) su funcionalidad.

3) Sobre sistema operativo GNU Linux

- a) Elija una distribución de GNU Linux y luego instale la misma en su equipo utilizando algún producto de virtualización. (Deberá investigar todo lo necesario para poder llevar a cabo la virtualización y resolver el resto de los problemas planteados a continuación).
- b) Describa en forma sintética que información encuentra en el directorio `/proc`, brinde una explicación general de los datos allí guardados. ¿Estos datos dejan de existir una vez apagado el equipo? ¿Por qué?
- c) Enumere y explique sintéticamente las llamadas al sistema que provee GNU Linux relacionadas con creación y manipulación de procesos (no más de 2 líneas por cada una).
- d) Describa las diferencias entre la utilización de las funciones: `fork()` y `exec()` de librería disponibles en lenguaje C en el sistema operativo GNU Linux.



PROYECTO N° 1

SISTEMAS OPERATIVOS

Licenciatura en Ciencias de la Computación - Segundo cuatrimestre de 2010
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



Para los ejercicios que se proponen a continuación deberá utilizar el compilador: gcc, utilizando el sistema operativo GNU Linux.

4) Creación de procesos

- a) Un proceso padre debe crear dos procesos hijos, cada proceso deberá indicar en un archivo (separado) si es el proceso: padre, primer hijo o segundo hijo. ¿En qué orden terminan? Realice las modificaciones necesarias para que se muestre en consola la identificación del proceso que termina.
- b) Un proceso debe crear dos hilos, tanto el proceso como los hilos deberán indicar en consola si es el proceso inicial, el primer hilo o el segundo hilo. ¿En qué orden terminan los hilos creados en el proceso?
- c) Un proceso padre debe crear tres procesos hijos:
 1. El padre crear un archivo denominado: `salida.txt` antes de invocar a los procesos hijos. El proceso padre debe esperar que los procesos hijos terminen su ejecución y al finalizar debe agregar en el archivo: `salida.txt` la leyenda “Se ha finalizado la actividad”.
 2. Cada proceso hijo debe invocar una imagen que realice la tarea asignada. Cada hijo debe realizar un ciclo de 10000 iteraciones, en el mismo debe mostrar por consola el texto “ID: X, Estoy realizando mi trabajo, PID: NNN” (Donde NNN es el número de proceso que le corresponde y X es la identificación del proceso: A, B o C). Además debe agregar en cada iteración el mismo texto en el archivo: `salida.txt`.
 3. ¿En qué orden terminan los procesos? Analice los resultados obtenidos durante la ejecución de los procesos creados.

5) Utilización de hilos

- a) Deberá implementar un programa que genere 5 hilos. Cada uno de ellos tendrá una letra identificatoria A, B, C, D y E. Además cada uno deberá ciclar por lo menos 10000 veces antes de finalizar.
 1. En el ciclo se deberá: mostrar en consola la letra identificatoria, la identificación del hilo y además realice un ciclo interno de 1000 repeticiones incrementando en 1 una variable interna a cada hilo.
 2. Después de crear los 5 hilos se debe crear un nuevo proceso, en el proceso hijo debe mostrar un mensaje que se ha creado correctamente. Muestre la identificación de cada hilo y del proceso para cada uno.

Observación: En todos los casos, el proceso que crea a los hilos debe esperar hasta que todos terminen. Analice los resultados obtenidos.



PROYECTO N° 1

SISTEMAS OPERATIVOS

Licenciatura en Ciencias de la Computación - Segundo cuatrimestre de 2010
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



- b) Deberá implementar dos programas que permitan mostrar en consola las siguientes secuencias de caracteres:

1. Secuencia: ABCDEABCDEABCDEABCDEABCDE...
2. Secuencia: A(B o C)(B o C)DEA(B o C)(B o C)DEA(B o C)(B o C)DE...
3. Secuencia: (A o B)CE(A o B)(A o B)DE(A o B)CE(A o B)(A o B)DE...

Observación: Luego de mostrar cada letra el programa debe esperar un segundo.

6) Comunicación entre procesos

- a) Explique brevemente el conjunto de funciones básicas que se utilizan para los segmentos de memoria compartida (Es decir: `shmget()`, `shmat()`, etc).
- b) Implemente dos programas que una vez en ejecución permitan que los procesos puedan enviarse entre sí (en forma sincrónica) mensajes individuales a través del recurso de memoria compartida entre ambos.
- c) Explique brevemente el conjunto de funciones básicas que se utilizan para la comunicación de procesos utilizando mensajes (Es decir: `msgsnd()`, `msgrcv()`, etc).
- d) Implemente dos programas que una vez en ejecución permitan que los procesos puedan enviarse entre sí (en forma sincrónica) mensajes individuales.
- e) Confeccionar un programa para resolver el problema de productor/consumidor. Deberá considerar que el *buffer* está compuesto por N elementos. Utilice los mecanismos de sincronización necesarios y que crea convenientes. Resolver el problema planteado utilizando:
 1. Un segmento de memoria compartida.
 2. Mensajes para la comunicación.

Observaciones generales:

- Deberán entregar para cada programa el archivo *Makefile* correspondiente, para poder efectuar la compilación y prueba del mismo (sin excepciones).
- En el código, entre eficiencia y claridad, se debe optar por la claridad. Toda decisión en este sentido debe constar en el informe que acompaña al programa implementado.
- Deben efectuar la programación utilizando técnicas de programación modular (incluyendo funciones y procedimientos), estudiadas durante la carrera.
- Se tendrán en cuenta aspectos relacionados con buenas técnicas de programación, simplicidad y modulación de los algoritmos implementados.
- El código entregado deberá estar comentado e indentado adecuadamente.



PROYECTO N° 1

SISTEMAS OPERATIVOS

Licenciatura en Ciencias de la Computación - Segundo cuatrimestre de 2010
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



- El modelo empleado en la resolución de cada problema debe ser adecuado a lo solicitado en cada ejercicio.

Documentación (informe):

- El informe en los ejercicios que así lo requieran deberá contener las explicaciones solicitadas.
- La documentación de cada uno de los programas implementados por cada ejercicio deberá estar orientada al programador, con los comentarios adecuados en todo el código.
- Debe entregar por cada programa el algoritmo en lenguaje de diseño de la resolución del problema/ejercicio, a partir del cual se hizo la programación en lenguaje C.
- En cada uno de los programas cada función deberá estar comentada, se deberá especificar el prototipo de la misma, explicando cada uno de los parámetros de entrada y/o salida. También deberá comentar para que utiliza cada variable, estructura de datos, recurso, etc.
- Para cada programa debe indicar forma de utilización del mismo en la línea de comandos (*shell*).

Condiciones de entrega:

- El proyecto debe ser realizado en comisiones de **2 alumnos sin excepciones**.
- El trabajo debe ser entregado el día 5 de noviembre de 2010 (en horario a definir).
- Cada programa en cada ejercicio deberá denominarse: **ejercicio<NI>.c**, donde *N* es el número de ejercicio, e *I* el número de inciso correspondiente.
- Deberá entregarse un folio de plástico correctamente cerrado que contenga:

☐ Un cdrom o pendrive con una carpeta llamada: `proyecto1`, la cual debe contener:

- Una carpeta `src`, con **subcarpetas** por cada ejercicio (por ejemplo: `/src/ejercicio3a/`)
- En cada una de las subcarpetas deberán estar presentes todos los archivos fuentes junto con cualquier otro archivo adicional que se necesite para poder compilar al código de cada ejercicio/inciso.

☐ El informe impreso (**con nombre y LU de cada uno de los alumnos**).

☐ El código impreso de todos los módulos implementados contenidos en la carpeta: `src`.

Condiciones de aprobación:

- No serán corregidos programas que generen errores durante la compilación.
- No serán corregidos los proyectos donde los archivos no respeten los nombres solicitados.
- No serán corregidos los proyectos donde en cada ejercicio no se encuentra el archivo *Makefile* correspondiente.
- En consecuencia: proyectos **no corregidos** serán desaprobados. ☹️

Aprobación de proyectos:

- El proyecto debe estar aprobado para el cursado de la materia.