

Blank sample to the class Operative Systems

Relevants:

Hector Robles Martinez (gh@hector290601)

Section:
Class Title



Facultad de Ingeniería
Ingeniería en computación
MMMM DD, YYYY
March 6, 2023

Contents

I	Warnings	2
1	Disclaimer	3
1.1	Info source	3
1.2	Limited Responsibility	3
1.3	Authors List	3
2	Death locks.	4
2.1	Mecanismos de detección	4
2.1.1	Algoritmo del avestruz	4
2.1.2	Modelo gráfico.	4
2.1.3	Análisis matricial.	5
2.1.4	Análisis de regiones de riesgo.	5
2.2	Prevencion de los deathlocks.	5
2.2.1	Exclusion mutua.	5
2.2.2	Hold & Wait	5
2.2.3	Noon-preemptive	5
2.3	Comunicación entre procesos.	5
2.3.1	Señales.	5
2.3.2	Pipes	5
3	Administración de la memoria.	7
II	Authors	8
4	Hector Robles Martínez	9
4.1	Attributions	9
4.2	First moddify	9
4.3	Last moddify	9
4.4	About me	9

Part I

Warnings

Chapter 1

Disclaimer

1.1 Info source

Toda la información contenida en éste documento, ha sido recolectada como notas de clase, sin propósito de ser un referente ni para su consulta externa, a menos que se indique lo contrario en ésta misma sección.

1.2 Limited Responsibility

Como se ha mencionado antes, la información contenida son partes de las notas de clase, cualquier error favor de comunicarse con el (o los) auto (o autores) del documento.

1.3 Authors List

Si usted hace cambios, modificaciones, o cualquier alteración al contenido de éste documento, favor de añadirse en la lista de autores.

Chapter 2

Death locks.

Bloqueo mutuo

Es un evento en el que dos o mas procesos se bloquean en espera de dos o mas recursos de modo que no pueden progresar independientemente del tiempo que pase..

Es como cuando dos camiones quieren dar vuelta y se bloquean mutuamente.

Tenemos un deadlock solo cuando se cumplen las siguientes 4 condiciones:

1. Exclusión mutua. Los procesos reclaman control exclusivo de los procesos.
2. Hold & Wait. Los procesos mantienen las asignaciones a los recursos en lo que esperan por los demas recursos que necesitan.
3. Non-preemption. La asignacion de los recursos no puede ser revocada.
4. Espera circular. Se genera una secuencia de procesos en espera de recursos asignados a otros procesos.

Estas condiciones sirven para saber cuando es un bloqueo 'normal' o cuando es un deathlock

2.1 Mecanismos de detección

2.1.1 Algoritmo del avestruz

En sistemas operativos y no implementa ningún mecanismo dejando a cargo del usuario el identificar bloqueos y tomar las medidas correctivas.

2.1.2 Modelo gráfico.

En este se dibuja un grafo en el que se pueden buscar los ciclos.

2.1.3 Análisis matricial.

Se pueden construir matrices que representen los grafos de asignacion de recursos que se operan para detectar ciclos.

Es costos en CPU y solamente permite detectar.

2.1.4 Análisis de regiones de riesgo.

Se define como región de riesgo el estado de una matriz de asignacion de recursos inmediata anterior al deathlock y se niegan las agnaciones de recursos que llevan a regiones de riesgo.

2.2 Prevencion de los deathlocks.

Para estos, basta negar alguna condicion del DL.

2.2.1 Exclusion mutua.

Reemplazar recursos preemptive por non-preemptive.

2.2.2 Hold & Wait

Podemos agrupar todos los recursos y asignarlos en una sola operaci6n

2.2.3 Noon-preemptive

Son operaciones que 'guardan' el estado inicial de los recursos y en caso de problemas preemptive, regeneran éste.

2.3 Comunicaci6n entre procesos.

2.3.1 Señales.

Estas permiten a un proceso de la jerarquia enviar una señal con un valor entero que interrumpe el flujo de la aplicaci6n para ejecutar la funci6n definida como handler.

2.3.2 Pipes

Todos los procesos se asocian a tres streams.

1. *STDIN* - Entrada estandar
2. *STDOUT* - Salida estandar.

3. *STDERR* - Salida de errores.

Proceso1 *> file.txt* Escribe la salida STDOUT a un archivo file.txt (sobreescribe el archivo)

Proceso1 *>> file.txt* Escribe la salida STDOUT a un archivo (append)

Proceso1 — **Proceso2** Asigna STDOUT de Proceso1 al STDIN de Proceso2

Proceso1 *2> file.txt* Escribe la salida STDERR de Proceso1 a file.txt

Chapter 3

Administración de la memoria.

Part II

Authors

Chapter 4

Hector Robles Martínez

Name Hector Robles Martínez

Mobile +52 5510604869

Mail robletes062901@gmail.com, hector.robles@daimler.com ,

Github [hector290601](https://github.com/hector290601)

4.1 Attributions

First update and attributions

4.2 First moddify

January 31, 2023

4.3 Last moddify

Febraury 1, 2023

4.4 About me

I'm a Computer Engenieering Student, actually I'm a passant on MBA.