

**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

<b>I</b>	<b>Warnings</b>	<b>2</b>
<b>1</b>	<b>Disclaimer</b>	<b>3</b>
1.1	Info source . . . . .	3
1.2	Limited Responsibility . . . . .	3
1.3	Authors List . . . . .	3
<b>2</b>	<b>Death locks.</b>	<b>4</b>
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
<b>II</b>	<b>Authors</b>	<b>6</b>
<b>3</b>	<b>Hector Robles Martínez</b>	<b>7</b>
3.1	Attributions . . . . .	7
3.2	First moddify . . . . .	7
3.3	Last moddify . . . . .	7
3.4	About me . . . . .	7

# 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 costoso 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 asignaciones 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 Hols & Wait**

Podemos agrupar todos los recursos y asignarlos en una sola operación

# Part II

# Authors

## Chapter 3

# Hector Robles Martínez

Name Hector Robles Martínez

Mobile +52 5510604869

Mail [robletes062901@gmail.com](mailto:robletes062901@gmail.com), [hector.robles@daimler.com](mailto:hector.robles@daimler.com) ,

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

### 3.1 Attributions

First update and attributions

### 3.2 First moddify

January 31, 2023

### 3.3 Last moddify

Febraury 1, 2023

### 3.4 About me

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