



Information Technology Institute



# Operating System Fundamentals



من اول الفيرشو اليزيشن مش جاي في الامتحان الي هي بعد المحاضره دي

لكن الديد لوك والميموري مانجمنت جاي في الامتحان

## Chapter Six

# DEADLOCKS

---

# Table of Content

- Introduction
- Deadlock Characterization
- Methods for Handling Deadlocks
- Deadlock Prevention
- Recovery from Deadlock

# INTRODUCTION

# The Deadlock Problem

لو عندي اثنين بروسيز واحد منهم بتقرا من الهارد ديسك وبتكتب على تيب درايف والتانيه هتقرا من التيب درايف وهتسبف من الهارد ديسك  
الوضع ده بنحاول منخلهوش يحصل او لو حصل نتلافاه باقل الخسائر لانه كل بروسيز عايز الاتنين ريسورسز في نفس الوقت لكن كل واحد ماسك ريسورسز

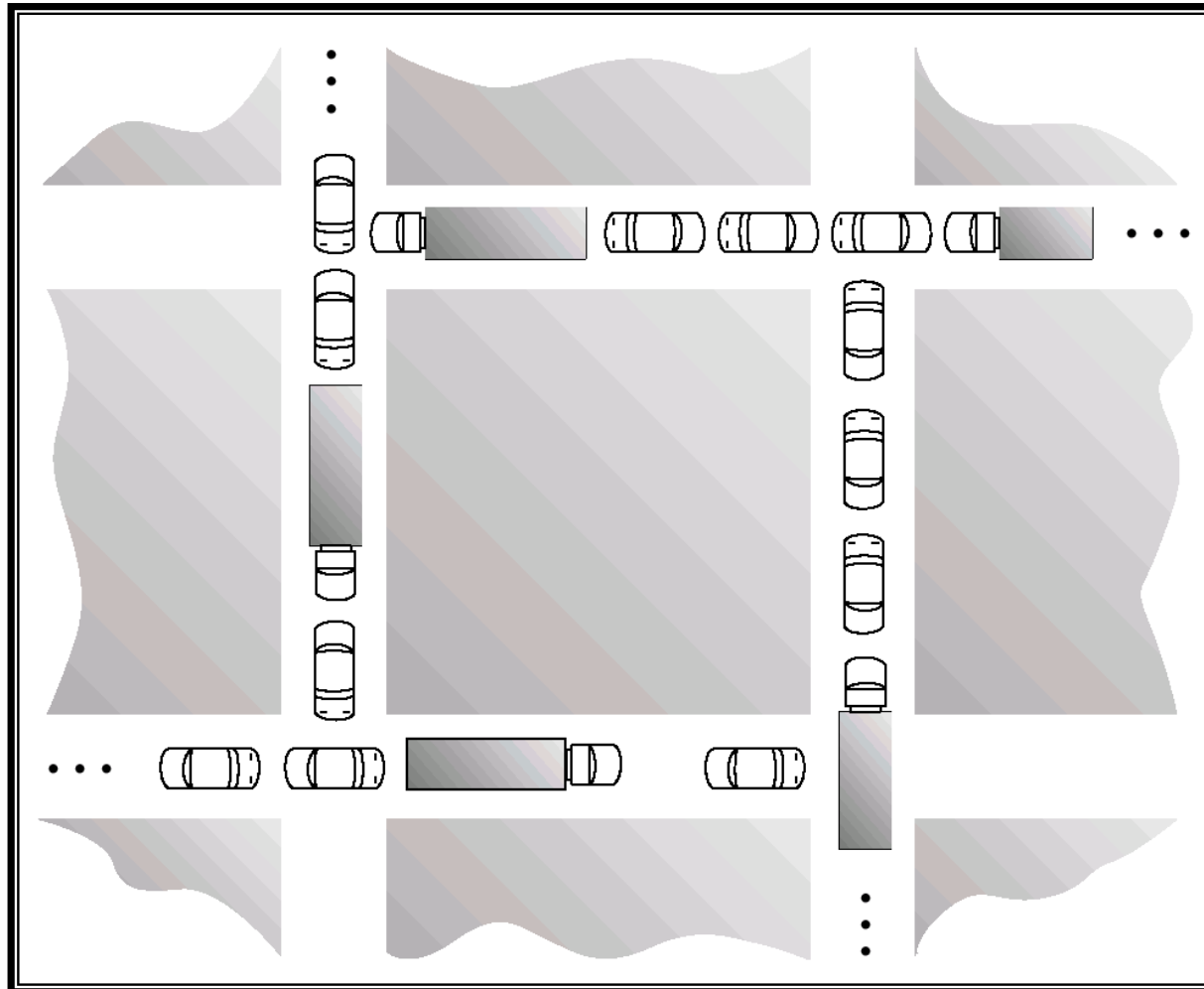
- A set of blocked processes each holding a resource and waiting to acquire a resource held by another process in the set. كده ولا هما هيتنفذو ولا هيسبو الريسورسز الي معاهم

- Example

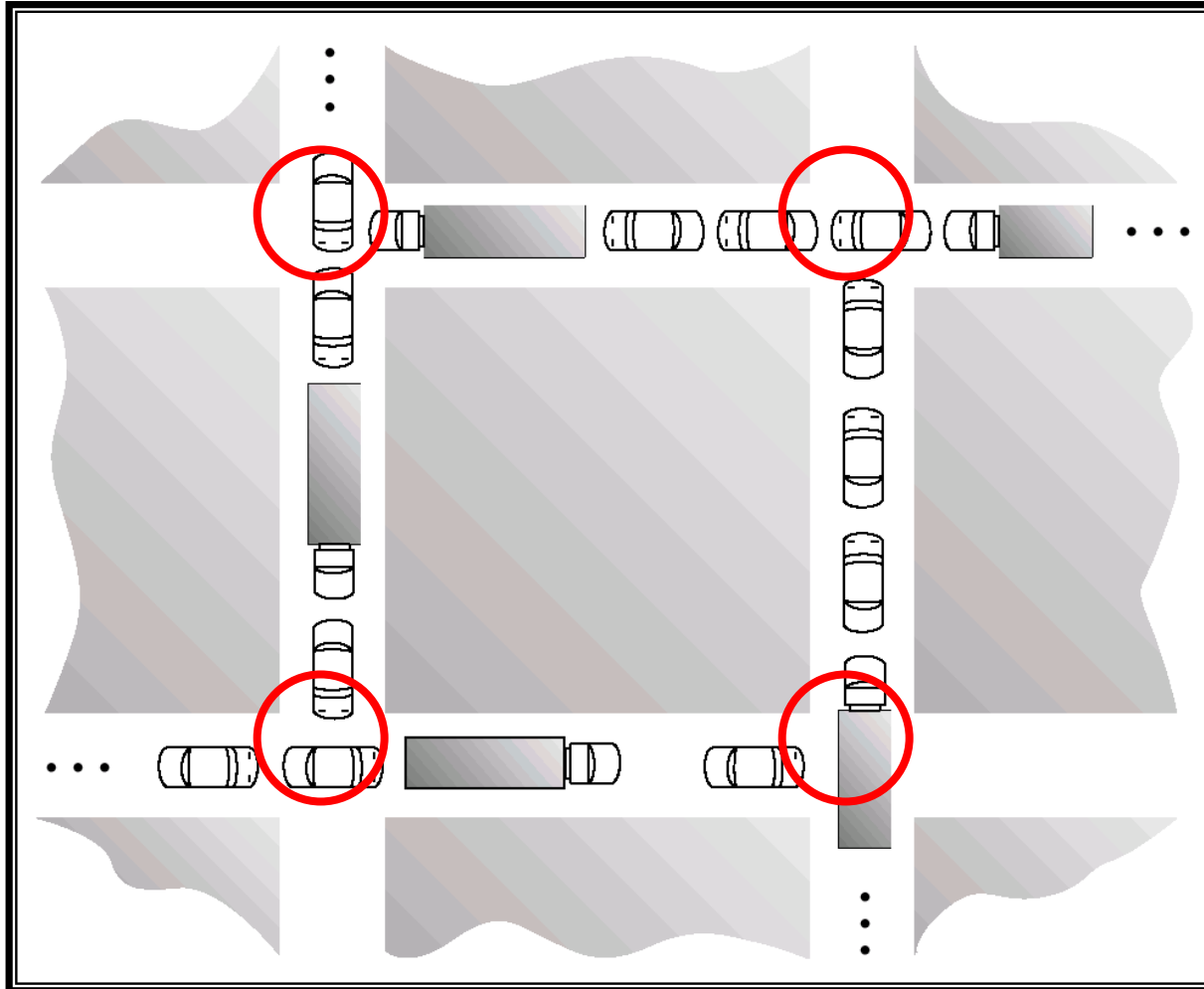
- System has 2 tape drives.
- P1 and P2 each hold one tape drive and each needs another one.
- semaphores A and B, initialized to 1

P1	P2
wait (A);	wait(B)
wait (B);	wait(A)

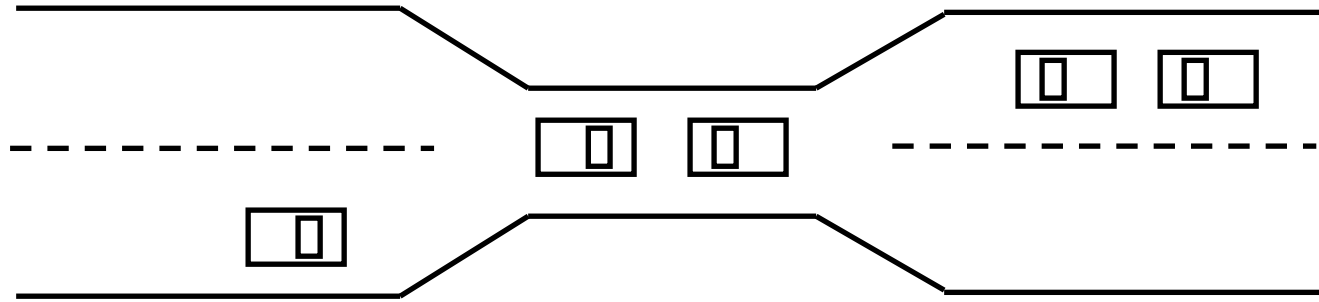
# Is this Deadlock?



# Yes, How to prevent it?



# Bridge Crossing Example



وعلشان افك المشكله دي لازم اعمل كيلنج لبروسيز ثم ارجعها تاني لكن لازم اتجنب الاسترفيشن ان انا اسيبها  
كثير مسالاش فيها

- Traffic only in one direction.
- Each section of a bridge can be viewed as a resource.
- If a deadlock occurs, it can be resolved if one car backs up (preempt resources and rollback).
- Several cars may have to be backed up if a deadlock occurs.
- Starvation is possible



# **DEADLOCK CHARACTERIZATION**

# Deadlock Characterization

ايه المصايب الي لما تكون متوفره بتوجد بيئه خصبه للديدلوك انه يحصل

مشاكل الاوبرينتج سيستم سمح انها تحصل

- **Deadlock** can arise if four conditions hold simultaneously.
  1. **Mutual exclusion**: only one process at a time can use a resource. ان يكون في بروسيس بتتمتع بخاصية الميوثوال ايكس كلوشن زي الطابعه مستحيل تطبع ورقتين في نفس الوقت.
  2. **Hold and wait**: a process holding at least one resource is waiting to acquire additional resources held by other processes. لو في بروسيس عايزه كذا ريسورس وماسكه بعض الريسورس دي وفي نفس الوقت في بروسيس ثانيه واقفه علي الريسورس الي الاولى مسكاها.
  3. **No preemption**: a resource can be released only voluntarily by the process holding it, after that process has completed its task. ان البروسيس ماسكه ريسورس معين ومش راضيه تسببه الا لما الريسورس الثانية الي هو عايزها تحيلها وتشتغل وبعدين تسبهم.
  4. **Circular wait**: there exists a set  $\{P_0, P_1, \dots, P_0\}$  of waiting processes such that  $P_0$  is waiting for a resource that is held by  $P_1$ ,  $P_1$  is waiting for a resource that is held by  $P_2$ , ...,  $P_{n-1}$  is waiting for a resource that is held by  $P_n$ , and  $P_n$  is waiting for a resource that is held by  $P_0$  ان كله محتاج كله وكله ماسك كله

هيبجي في الامتحان دياجرام وتقولي فيه ديدلود وتحله ازاي

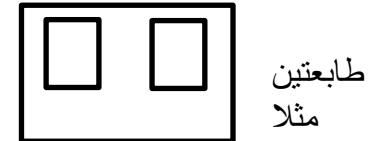
# Resource-Allocation Graph

تعالى نشوف الديجرام الي هيجي زيه في الامتحان نرسمه ازاي ونعرف منه في ديدلوك ولا لا

- **Process** 

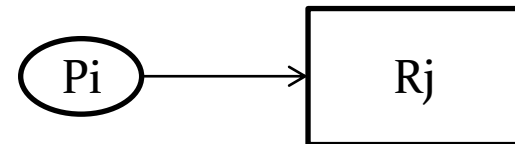
لو الريسورس فيه انستانس واحد يبقى مربع واحد

- **Resource Type with 2 instances**



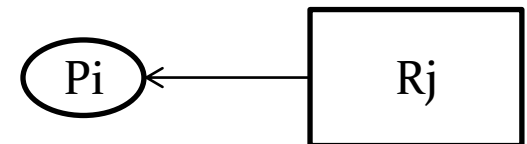
لما البروسيس تطلب الريسورس هنطلع سهم

- **$P_i$  requests instance of  $R_j$**



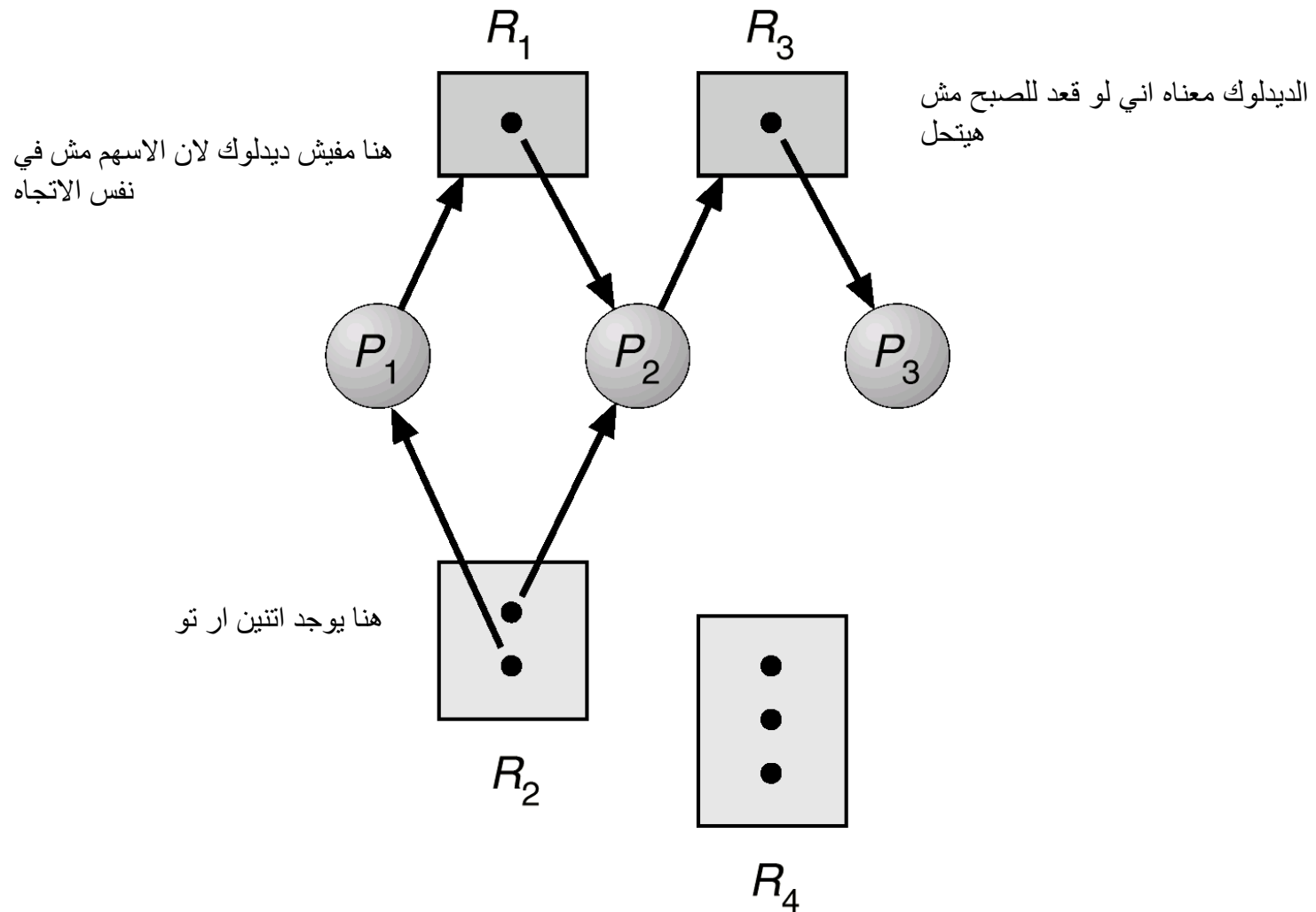
لما تكون البروسيس ماسك الريسورس

- **$P_i$  is holding an instance of  $R_j$**

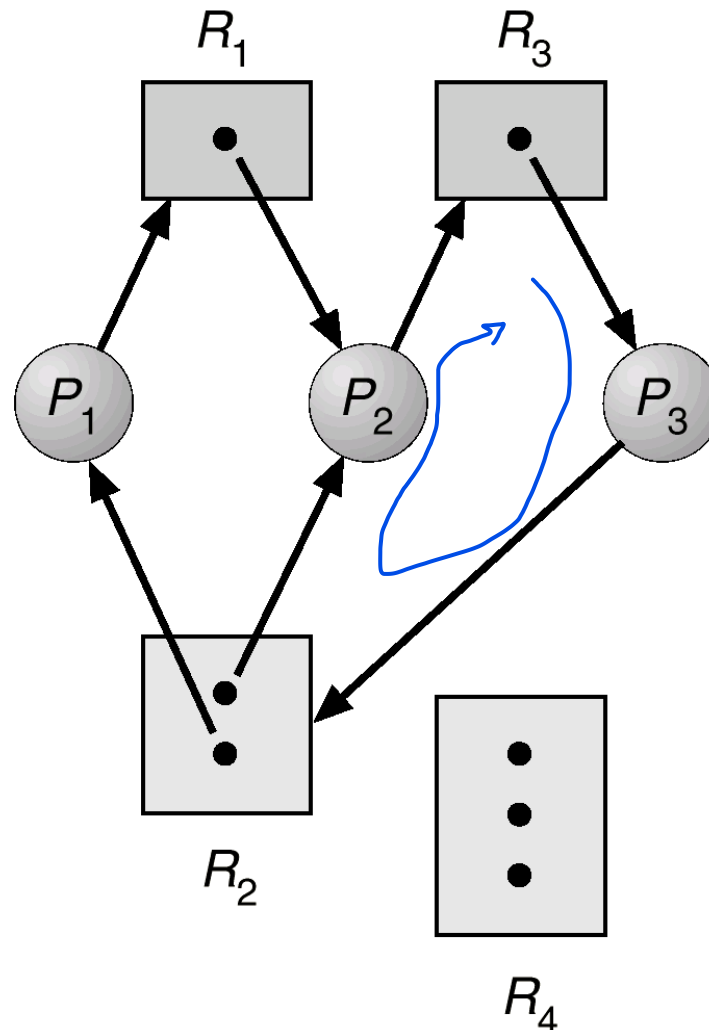


هندور على سهم لافف في اتجاه واحد قافل دايره لما نلاقه بيقى فيه احتمال ان في ديدلوك

# Example of a Resource Allocation Graph



# Resource Allocation Graph With A Deadlock



ديدلوك

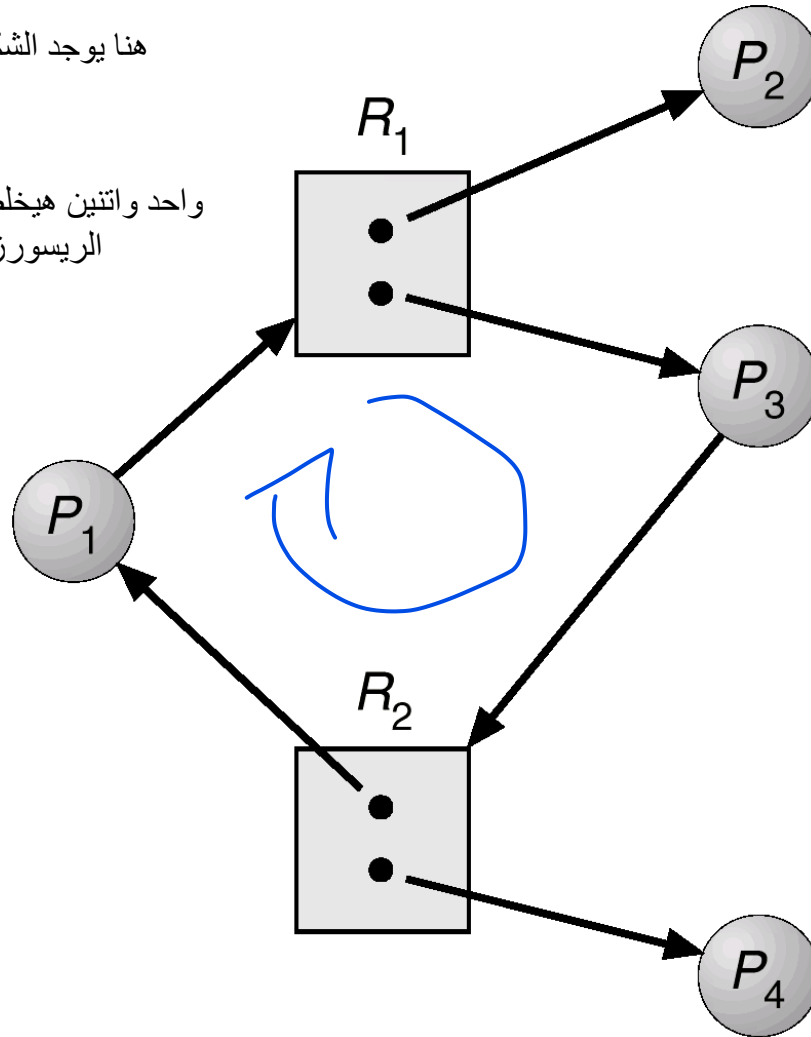


لو في اكثر من ريسورس يقلل احتمال  
وجود ديدلوك حتى لو يوجد مغلق

# Resource Allocation Graph With A Cycle But No Deadlock

هنا يوجد الشكل المغلق لكن  
مفيش ديدلوك

واحد واثنين هيخلصو ويسيبو  
الريسورز الى معاها



# Basic Facts

- If graph contains no cycles no deadlock.
- If graph contains a cycle
  - if only one instance per resource type, then deadlock.
  - if several instances per resource type, possibility of deadlock. احتمال یوجد او لا

# **METHODS FOR HANDLING DEADLOCKS**



# Methods for Handling Deadlocks

ازاي بهندل الديدلوک

مفیش اوبریتنج سیستم هیتجنب الیدلوک علی طول قبل حدوثه

- Ensure that the system will *never* enter a deadlock state.
- Allow the system to enter a deadlock state and then recover. ویاریت ریکوفر باقل الخسائر وفي اقل وقت
- Ignore the problem and pretend that deadlocks never occur in the system; used by most operating systems.

هنا الاوبریتنج سیستم یهمل المشكله خالص لحد ما الراجل الي قاعد علی  
الاوبریتنج سیستم یعمل ریستور فکل الریسورس هتتساب وکل  
البروسیسس هیحصلها کلینج والیدلوک هیتفک من نفسه وده احد الحلول بدل  
ما انا ک اوبریتنج سیستم اقتل بروسیس منهم واقعد افکر اقتل مین

# **DEADLOCK PREVENTION**

# Deadlock Prevention

الاربع مصايب الي كانو متجمعين وببسيبو الديد لوك ف عايزين نمنعهم من الاول

الميوشول احيانا بيبقى ف ايدي لو الريسورس ينفذ نشاركه ببقى تمام لكا حاجه زي الطابعه عمرها ما تطبع ورقتين في نفس الوقت لكن حاجه زي الهارد ينفذ اخزن في كذا مكان عادي

- **Mutual Exclusion**

- Not required for sharable resources; must hold for non-sharable resources.

- **Hold and Wait** هنا الاوبريتج سيستم ميخلش بروسيس تقوم اصلا الا لما يتوفر لها الريسورس الي عايز اهاو هنعمل حاجه ثانيه ان مفي بروسيس تطلب اي ريسورس وهي ماسكه ريسورس ثاني

- must guarantee that whenever a process requests a resource, it does not hold any other resources. ✓
  - Require process to request and be allocated all its resources before it begins execution, or allow process to request resources only when the process has none.
  - Low resource utilization; starvation possible.

هنا الاسترفيشن محتمل جده لان افرض بروسيس عايزه خمس ريسورس ف هتستنى الخمسه دول وقت كبير لما يفضو و هيكون فيه لوو ريسورس يوتيليزيشن لان الريسورس كده هستعملها قليل

# Deadlock Prevention Cont'd

لو بريسوس عايزه ريسورس من برويسس ثاني طب ما عادي البرويسس الثاني دى تسييه عقبال لما البرويسس الاول يخلص وبعدين يرجعه ثاني

- **No Preemption**

- If a process that is holding some resources requests another resource that cannot be immediately allocated to it, then all resources currently being held are released.
- Preempted resources are added to the list of resources for which the process is waiting.
- Process will be restarted only when it can regain its old resources, as well as the new ones that it is requesting.

- **Circular Wait** هنا هنرقم الريسورسس ونديها ترتيب لكل برويسس كده يبقى كل واحد ليه دور وحتجنب الديد لوك

- Impose a total ordering of all resource types, and require that each process requests resources in an increasing order of enumeration.

طب لو انا اخدت كل الاحتياطات الي فاتت دي وحصل برك ديدلوك

# RECOVERY FROM DEADLOCK

# Recovery from Deadlock

الحل الاول ده وهو الحل القاسي وهو اني اقتل كل البوسيسس واعاقبهم كلهم واني اقتلهم يعني اني اسيف حالتهم ويسببو كل الريسورسس الي معاهم ولما ارجع هبدأ من حيث انتهو

- Abort all deadlocked processes.

الحل الثاني ده هبتدى اقتل بروسيس واحد واحد بس مش عشوائي طبعا لحد ما الديدلوك تتفتح

- Abort one process at a time until the deadlock cycle is eliminated.

على اي اساس هختار البروسيسس الي هقتلها دي افكار ممكن نتبع احد منها على حسب كل شخص

- In which order should we choose to abort?

- Priority of the process. واحد يقولك هقتل الاقل الاهميه
- How long process has computed, and how much longer to completion. هنا حسب البروسيسس فات فيها قد ايه واقتل الي فات فيها قليل
- Resources the process has used. واحد ثاني يقولك هقتل البروسيسس الي محتاجه اقل ريسورس
- Resources process needs to complete. هبقى سهل اعملها ثاني
- How many processes will need to be terminated. الي محتاجه ريسورس كثير
- Is process interactive or batch? هحسب
- لو قتلت الباتش اليوسر مش هيحس

الى جهاز اللى صلاحي لازم

# Recovery from Deadlock: Resource Preemption

- **Selecting a victim** البروسيس الي هتقتل وتسبب اقل تكلفه
  - minimize cost.
- **Rollback** البروسيس الي اتقتلت هرجع اشغها من حيث انتهينا
  - return to some safe state, restart process for that state.
- **Starvation** مش اقتل نفس البروسيس كل مره
  - same process may always be picked as victim, include number of rollback in cost factor.



المسأله الي هتيجي هي قولتي كلام عندك بروسيس واحد محتاج ريورس واحد واثنتين و بي تو محتاجه ار 3 وار 4 المطلوب الاول ارسم الجراف ثم حددي اذا كان في سايكل ولا لا وبعدين هل السايكل دي ديدلوك ثم اكتبلي بالكلام ازاي الديدلوك ده هيجصل

الوحيد الي مش جايه في الامتحان هو الفيرشواليزيشن