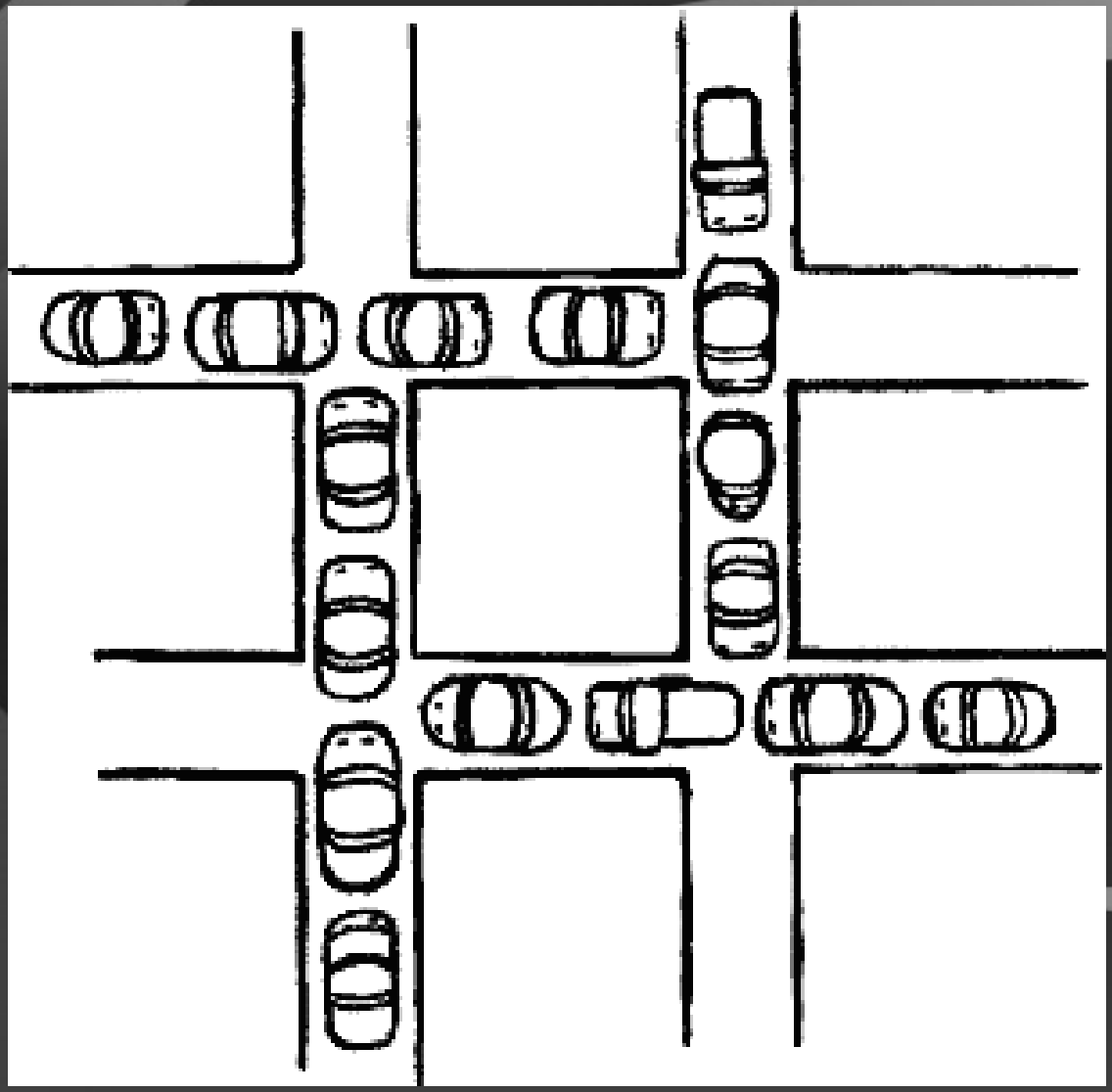
Konkurensi: Deadlock dan Starvation

Pertemuan 6

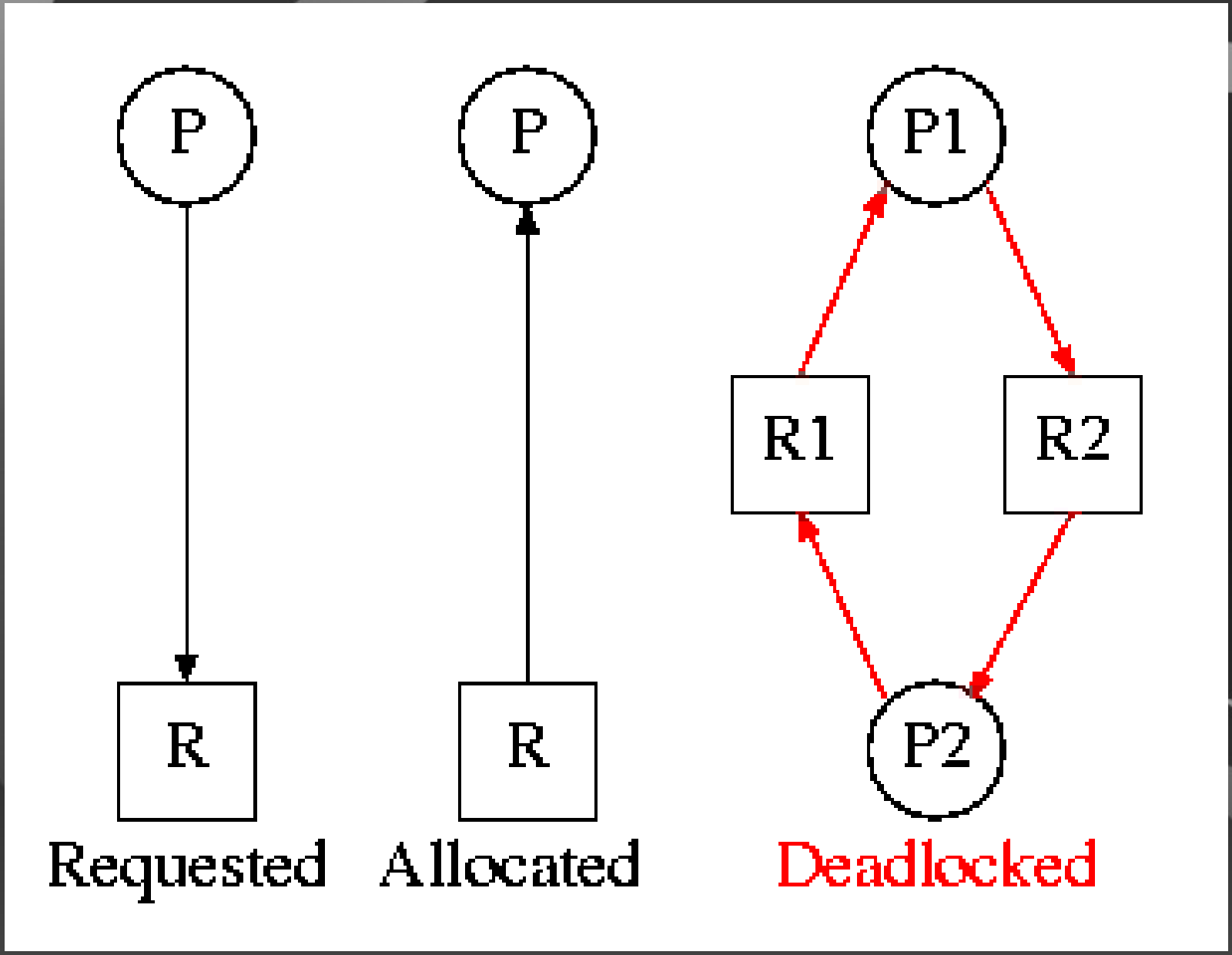
DEADLOCK

- Permanent blocking dari sekumpulan proses yang saling bersaing mendapatkan sumber daya sistem atau komunikasi
- Tidak ada solusi yang efisien
- Bisa dikatakan proses saling menutupi satu sama lian

Ilustrasi



Ilustrasi #2



Ilustrasi #3

COP: Thread #1 demands Resource #2 but Criminal owns the LOCK

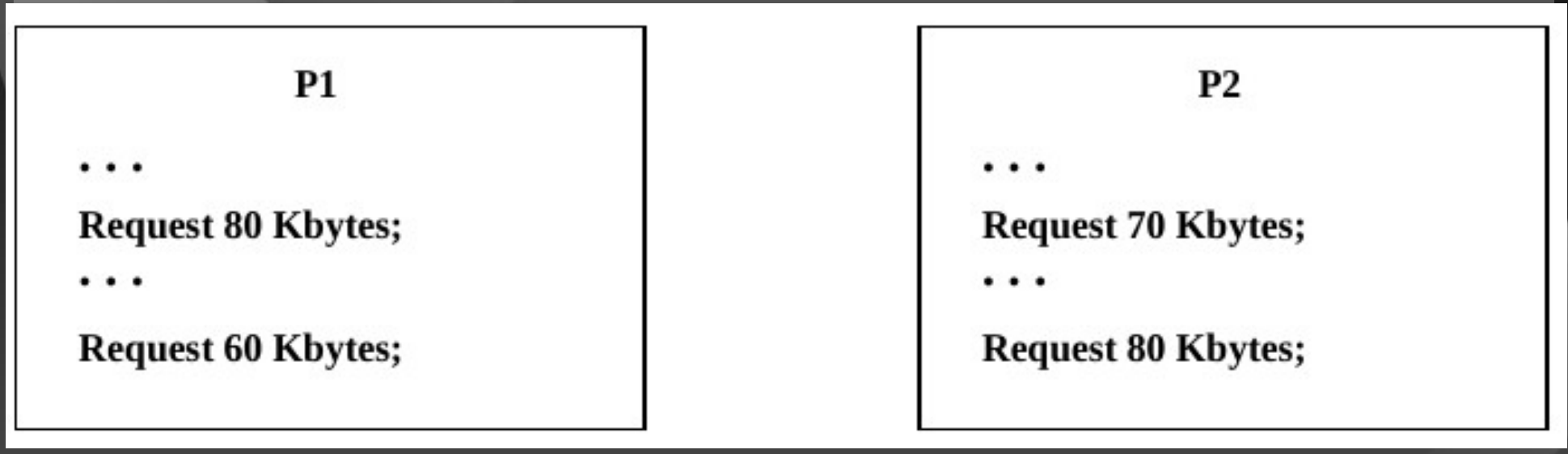
CRIMINAL: Thread #2 demands Resource #1 but Cop owns the LOCK

CRIMINALS FRIEND: Resource #2, the owner of the LOCK is Cop

HOSTAGE OF CRIMINAL: Resource #1, the owner of the LOCK is CRIMINAL

Lanjutan

- Deadlock terjadi jika setiap proses memegang satu sumber daya dan meminta sumber daya lain
- Space (ruang di memory) tersedia bagi alokasi 200 KB, dan deretan kejadian berikut terjadi
- Deadlock terjadi jika kedua proses bergerak ke permintaan kedua mereka



Kondisi Untuk Deadlock

- Mutual exclusion
  - Hanya satu proses yang boleh menggunakan suatu sumber daya pada satu waktu
- Hold-and-wait
  - Suatu proses boleh memegang sumber daya yang dialokasikan selama menunggu assignment yang lain

Lanjutan

- No preemption
  - Tidak ada sumber daya yang dapat dipaksa dihilangkan dari suatu proses yang memegangnya
- Circular wait
  - Adanya rantai tertutup (closed-chain) proses-proses, sehingga setiap proses memegang setidaknya satu sumber daya yang diperlukan oleh proses berikutnya dalam rantai tersebut

Pencegahan Deadlock

- Mutual Exclusion
  - Harus didukung oleh SO
- Hold and Wait
  - Mengharuskan suatu proses meminta (request) semua sumber daya yang dibutuhkannya pada satu waktu

Lanjutan

- No Preemption
  - Proses harus melepaskan sumber daya dan merequest lagi
  - SO boleh men-preempt suatu proses untuk mengharuskannya melepas sumber dayanya
- Circular Wait
  - Definisikan suatu pengurutan linier dari jenis-jenis sumber daya

Deadlock Avoidance

- Suatu keputusan dibuat secara dinamis apakah permintaan alokasi sumber daya terkini akan, jika diijinkan, secara potential mengarah ke deadlock
- Memerlukan pengetahuan permintaan proses mendatang

Lanjutan

- Kebutuhan resource maksimum harus dinyatakan sebelumnya
- Proses di bawah konsiderasi harus bersifat independen; tidak ada kebutuhan sinkronisasi
- Harus ada sejumlah fix sumber daya yang akan dialokasikan
- Tidak ada proses yang boleh exit selama memegang resource

What to Do

- Jangan mulai suatu proses jika tuntutannya dapat mengarah ke deadlock
- Jangan ijinikan suatu permintaan sumber daya berikutnya (incremental) untuk suatu proses jika alokasi ini dapat mengarah ke deadlock

Jika Terjadi Deadlock

- Batalkan semua proses terdeadlock
- Back up setiap proses terdeadlock ke beberapa checkpoint terdefinisi sebelumnya & restart semua proses
  - Mungkin terjadi original deadlock

Lanjutan

- Berikutnya batalkan proses terdeadlock sampai deadlock tidak ada
- Kemudian preempt sumber daya sampai deadlock hilang (habis)

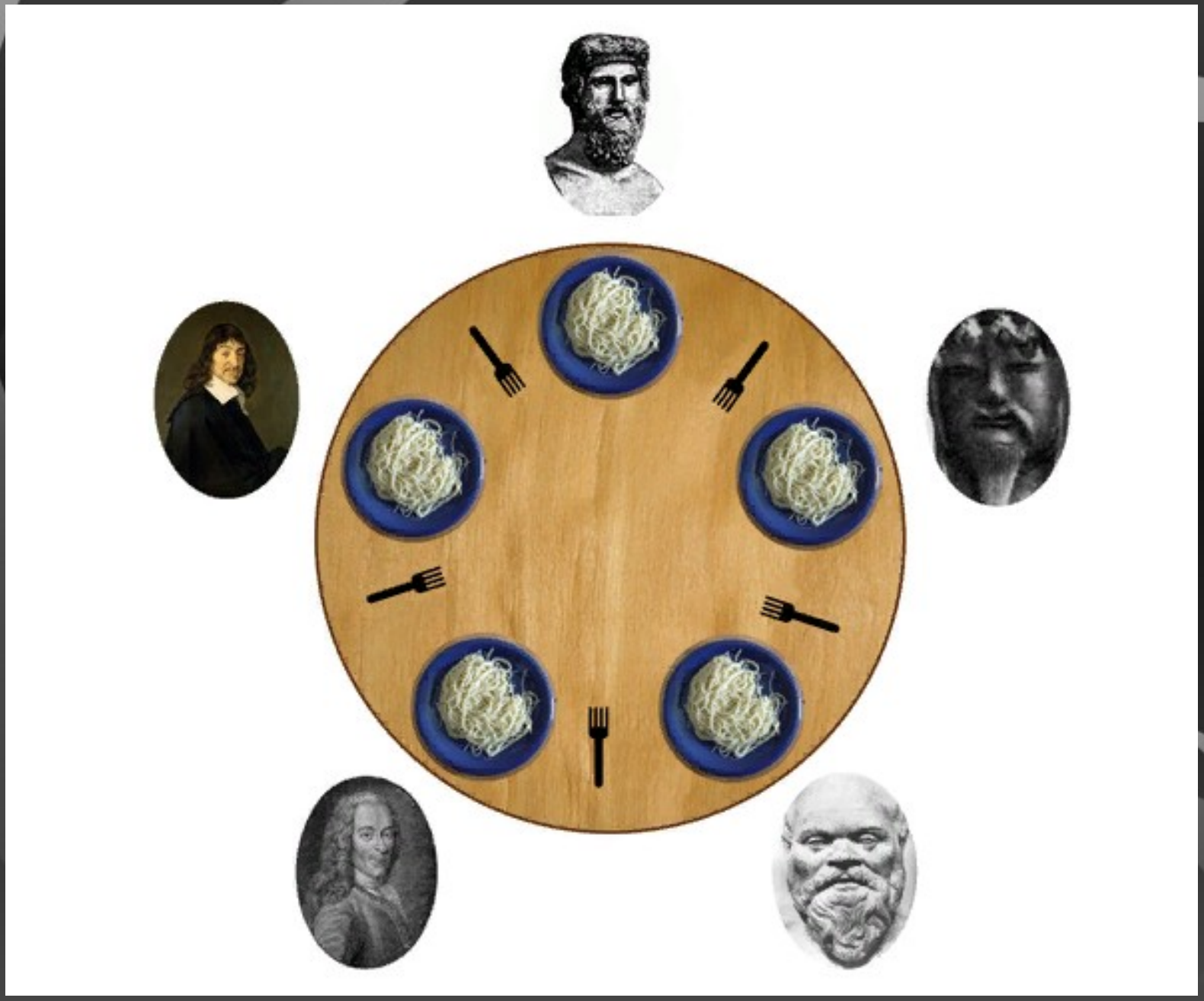
Starvation

- Sebuah masalah yang terjadi di dunia komputing yang di mana sebuah proses secara terus menerus menolak sumber daya yang dibutuhkan untuk diproses
- Starvation → Deadlock

Jenis Starvation

- Ada berbagai macam jenis dari Starvation, seperti Starvation akan sumber daya maupun akan CPU. Ada banyak sekali contoh akan starvation. Sebagai contohnya adalah Dini Philosophes Problem

Ilustrasi



Starvation vs Deadlock

- **Process:**
- Dalam deadlock, 2 athread atau proses akan menunggu satu sama lain.
- Dalam starvation, ketika dua atau lebih thread atau proses menunggu sumber yang sama, satu akan mundur dan membiarkan proses lain berjalan terlebih dahulu dan proses yang mengalami starvation akan mencoba lagi. Oleh karena itu semua semua proses atau thread akan berlanjut.

Starvation vs Deadlock

- **Rolling Back:**
- Dalam Deadlock, semua proses dengan prioritas tinggi maupun prioritas rendah akan menunggu satu sama lain tanpa henti
- Tapi dalam Starvation, proses prioritas rendah yang akan menunggu atau mundur, sedangkan prioritas tinggi tetap berjalan

Starvation vs Deadlock

- **Waiting or Lock:**
- Deadlock adalah antrian memutar
- Dan starvation adalah sejenis livelock yang terkadang dapat membantu satu sama lain agar tidak terjadi deadlock

Starvation vs Deadlock

- **Deadlock and Starvation:**
- Deadlock dapat menyebabkan starvation, tetapi Starvation tidak menyebabkan Deadlock
- **Penyebab**:
- Sebuah Deadlock akan terjadi dikarenakan oleh mutual exclusion, hold and wait, no preemption atau antrian memutar
- Starvation terjadi karena kelangkaan sumber daya, managemen sumber daya yang tidak terkontrol, dan prioritas proses

Mekanism Konkurensi UNIX

- Pipes
- Messages
- Shared memory
- Semaphores
- Signals

Mekanism Konkurensi UNIX

- Menyertakan semua mekanisme yang terdapat pada UNIX
- Operasi atomic berjalan tanpa interupsi dan tanpa interferensi

Sinyal Proses UNIX

Value	Name	Description
01	SIGHUP	Hang up; sent to process when kernel assumes that the user of that process is doing no useful work
02	SIGINT	Interrupt
03	SIGQUIT	Quit; sent by user to induce halting of process and production of core dump
04	SIGILL	Illegal instruction
05	SIGTRAP	Trace trap; triggers the execution of code for process tracing
06	SIGIOT	IOT instruction
07	SIGEMT	EMT instruction
08	SIGFPE	Floating-point exception
09	SIGKILL	Kill; terminate process
10	SIGBUS	Bus error
11	SIGSEGV	Segmentation violation; process attempts to access location outside its virtual address space
12	SIGSYS	Bad argument to system call
13	SIGPIPE	Write on a pipe that has no readers attached to it
14	SIGALRM	Alarm clock; issued when a process wishes to receive a signal after a period of time
15	SIGTERM	Software termination
16	SIGUSR1	User-defined signal 1
17	SIGUSR2	User-defined signal 2
18	SIGCHLD	Death of a child
19	SIGPWR	Power failure

Spinlock & Semaphore

- Spinlocks dan Semaphore adalah dua mekanisme yang digunakan untuk melindungi critical section di dalam kernel
- Spinlock memastikan hanya ada satu thread yang memasuki critical section dalam satu waktu. Thread lain yang ingin masuk critical section harus menunggu hingga thread pertama keluar