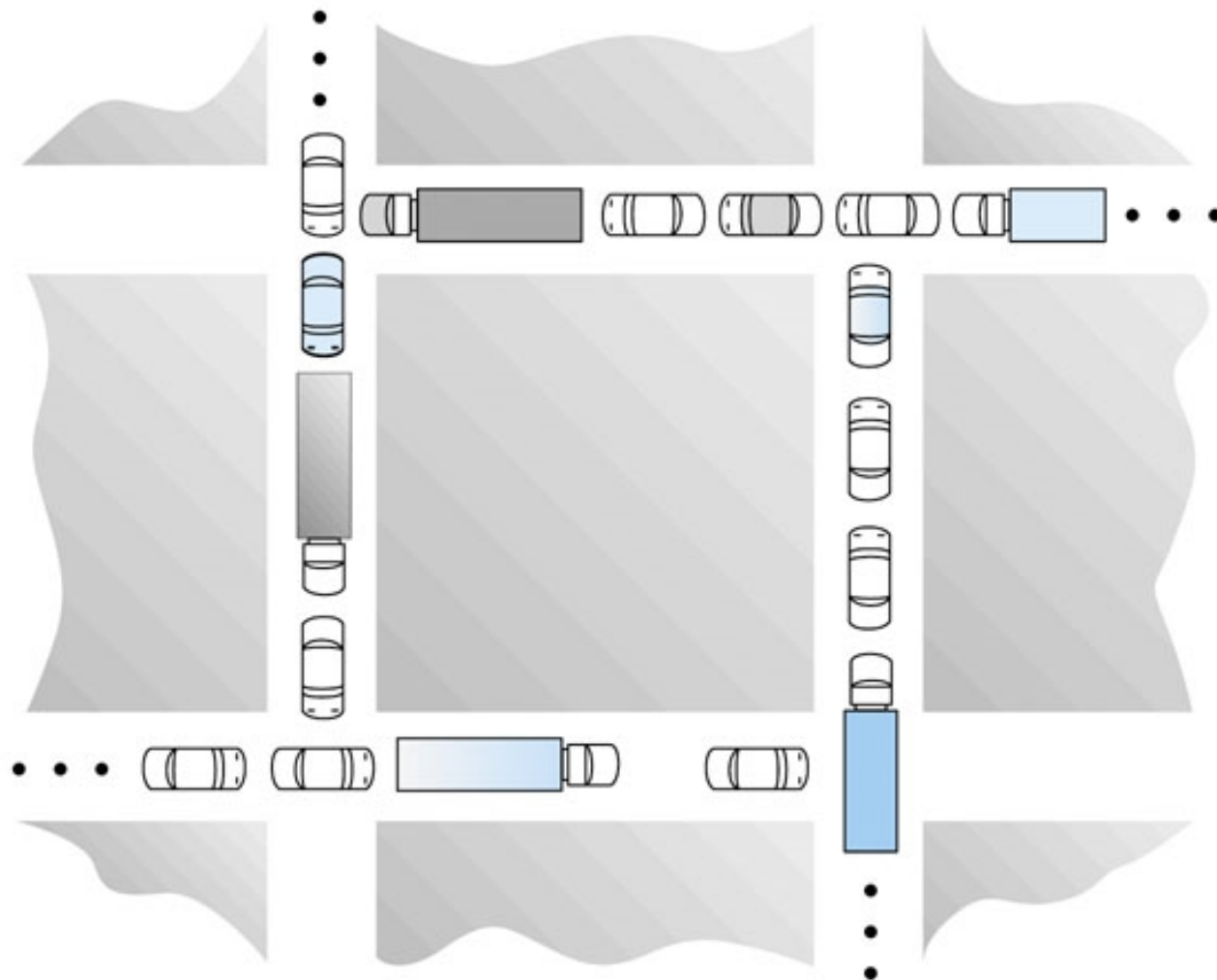


# Sistemas Operacionais I

Deadlocks

Prof. Leandro Marzulo

# Exemplo



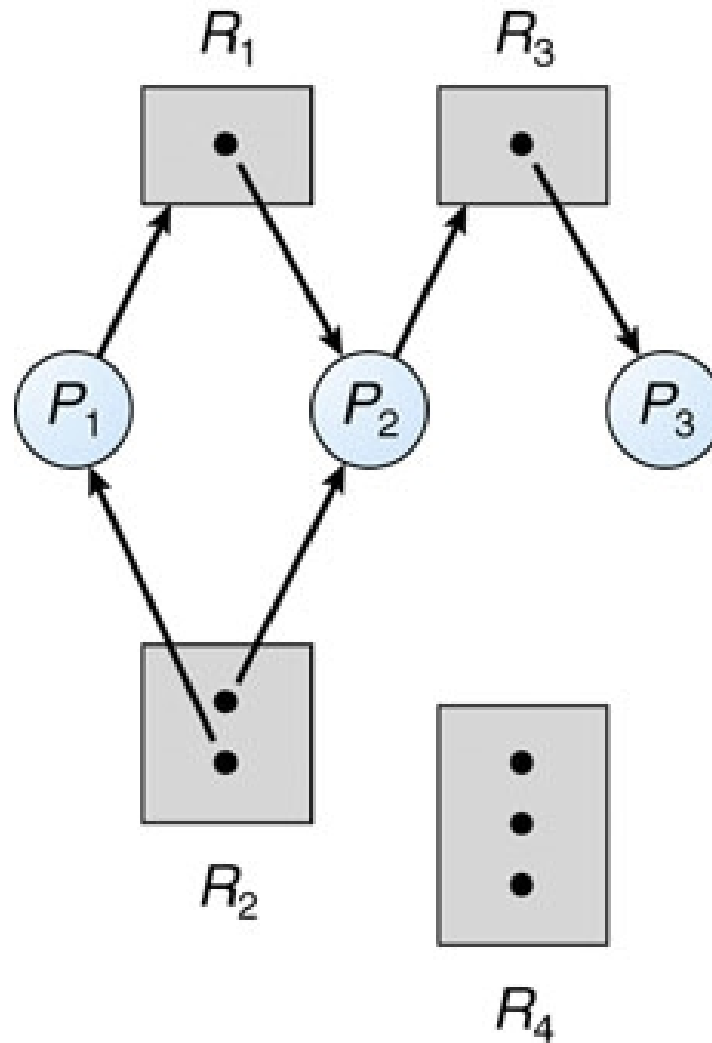
# Condições

- Um deadlock ocorre quando as seguintes condições ocorrem simultaneamente em um sistema:
  - Exclusão mútua – Pelo menos um recurso deve estar alocado em modo não-compartilhável (um processo por vez usa o recurso).
  - Posse e espera – Um processo deve estar de posse de pelo menos um recurso e esperando para adquirir recursos adicionais que, no momento, estejam sendo mantido por outros processos.
  - Inexistência de preempção de recursos – Recursos não podem ser interceptados, isto é, um recurso só pode ser liberado voluntariamente pelo processo que o detém.
  - Espera circular – Deve existir um conjunto  $\{P_0, P_1, \dots, P_n\}$  de processos em espera de tal modo que  $P_0$  esteja esperando por um recurso alocado a  $P_1$ ,  $P_1$  esteja esperando por um recurso alocado a  $P_2$ , ...,  $P_{n-1}$  esteja esperando por um recurso alocado a  $P_n$  e  $P_n$  esteja esperando por um recurso alocado a  $P_0$ .

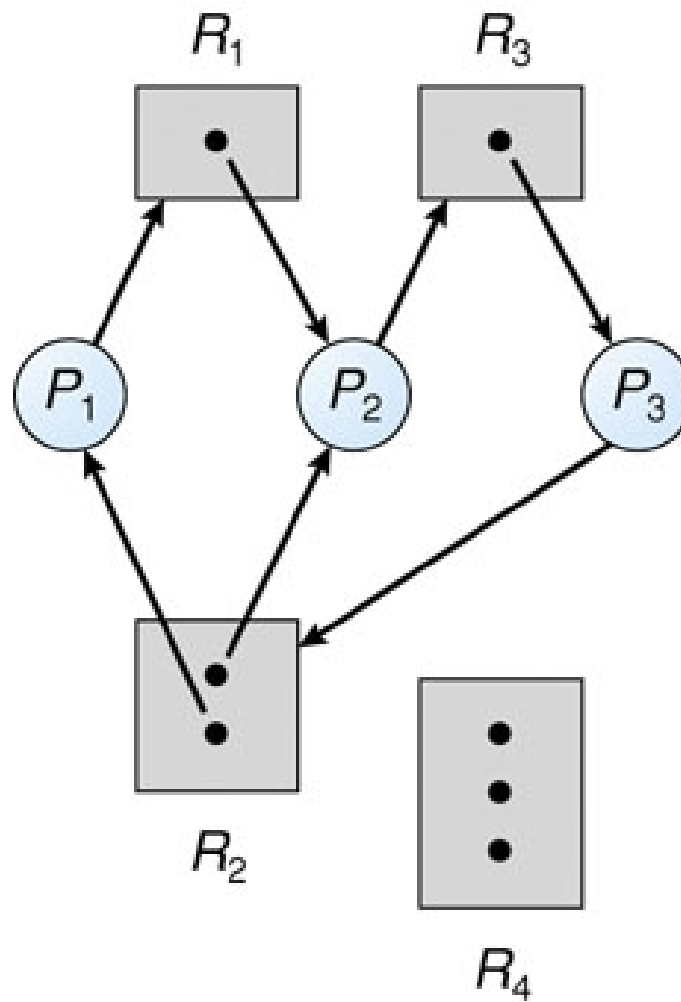
# Código com deadlock

- EXEMPLO da pg 149 do livro.

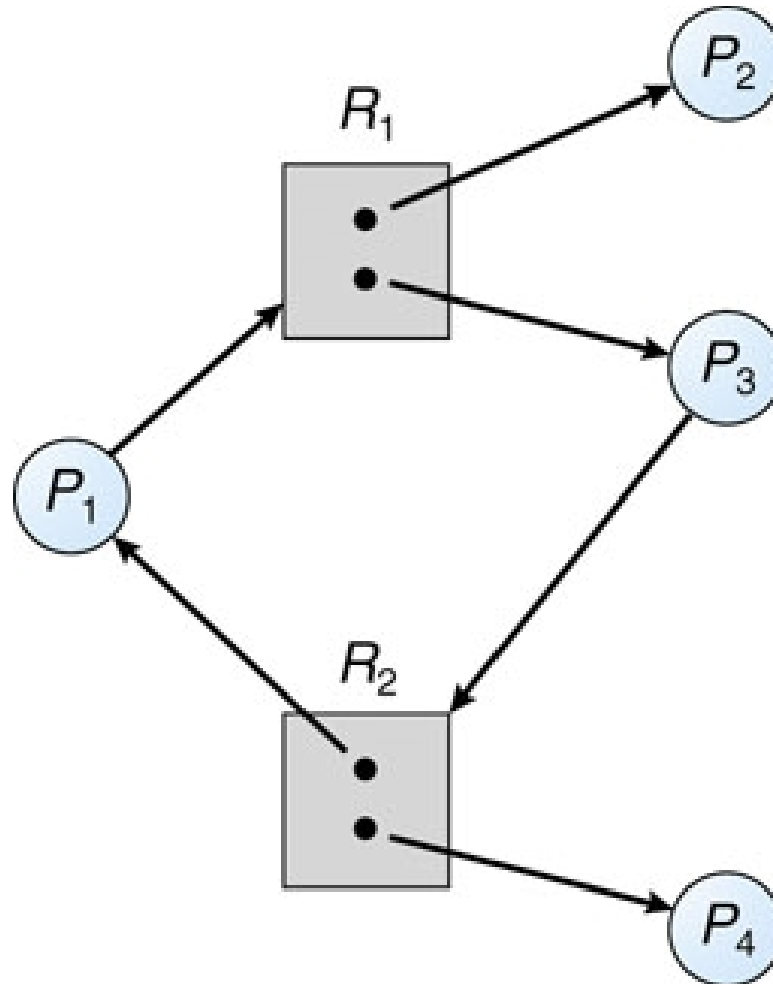
# Grafo de alocação de Recursos



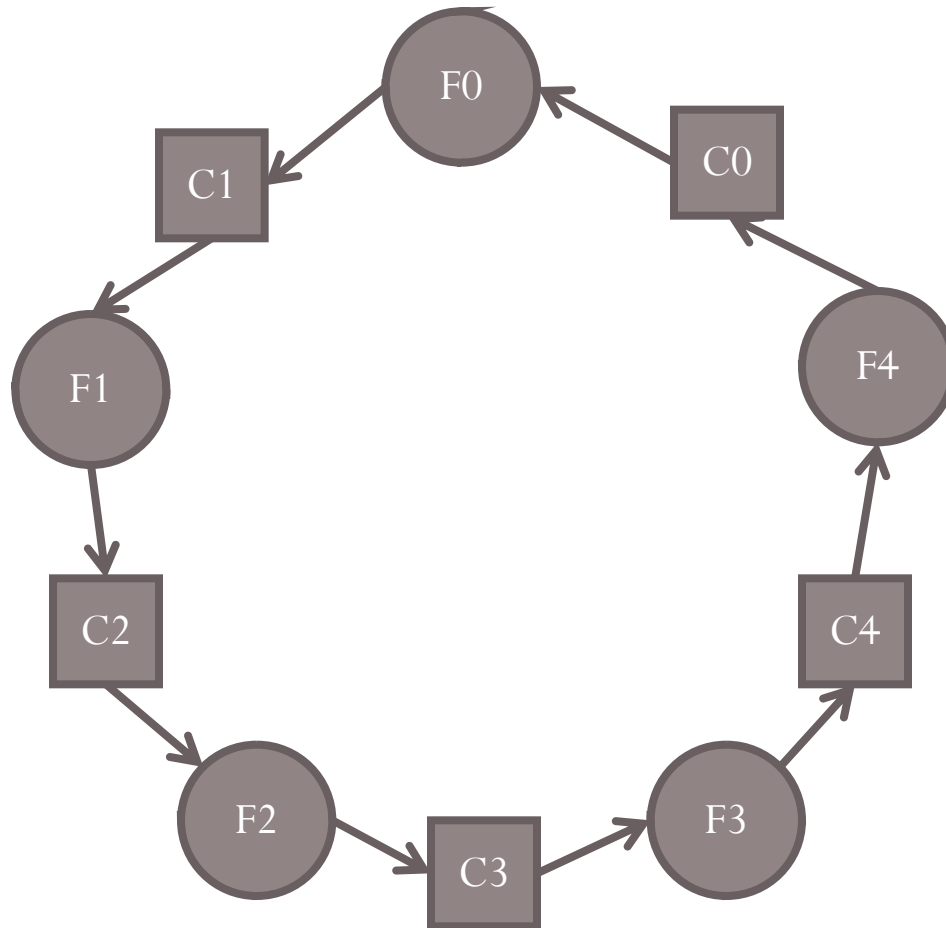
# Deadlock



# Sem Deadlock



# Deadlock no problema dos Filósofos





# Métodos para manipulação de Deadlocks

- Garantir que deadlocks não ocorram
  - Prevenir — garantir que 1 das 4 condições jamais ocorram
  - Impedir — SO recebe antecipadamente informações adicionais relacionadas a que recursos um processo solicitará e usará durante seu tempo de vida. Com esse conhecimento, ele pode decidir, para cada solicitação, se o processo deve ou não esperar, considerando os recursos disponíveis no momento, os recursos alocados concorrentemente e as futuras solicitações e liberações de cada processo.
- Detectar e recuperar
- Ignorar (mais comum)
  - Windows
  - Linux
  - Unix
  - Responsabilidade do programador

# Prevenção

- Exclusão mútua
  - Alguns recursos podem ser compartilháveis (arquivos de leitura)
  - Em geral não podemos prevenir esta condição
- Posse e espera
  - Protocolo – Cada processo solicita e recebe todos os recursos antes de usá-los – precedência na execução das chamadas de sistema de alocação de recursos.
  - Alternativa – Não pode solicitar outros recursos enquanto usa outros (deve liberar antes) – solicitar em blocos.
  - Utilização baixa dos recursos e inanição.

# Prevenção

- Inexistência de Preempção
  - Se um processo tem recursos e vai esperar por algum outro ele libera todos (e passa a esperar por todos).
  - Se um processo solicita recursos, verificamos se estão disponíveis
    - Se estiverem, serão alocados
    - Se não, verificamos se foram alocados a outro processo que esteja esperando por recursos adicionais e liberamos os recursos necessários
    - Se estão indisponíveis mas não estão alocados a nenhum processo, o processo espera e, enquanto isto, seus recursos podem ser interceptados. O processo só volta a executar quando obtiver todos os recursos.
  - Aplicado a recursos cujo estado pode ser salvo (contexto) como Banco de registradores e Memória (não impressora)

# Prevenção

- Espera Circular
  - Numerar os recursos
  - Obter locks na ordem
  - Programador
  - Bibliotecas com verificação