

---

# MAC5753 - Sistemas Operacionais

Daniel Macêdo Batista

IME - USP, 17 de Setembro de 2020

O problema da  
seção crítica

---

Semáforos

---

## O problema da seção crítica

## Semáforos

▷ O problema da  
seção crítica

---

Semáforos

---

# O problema da seção crítica

# Quais são as ações atômicas?

O problema da  
seção crítica

Semáforos

- Ação atômica realiza uma transformação de estado (mudança de variáveis) de forma indivisível
- Estados intermediários não podem ser vistos por outros processos
- Depende do hardware
  - Atribuições de variáveis são atômicas

# Exemplo

O problema da  
seção crítica

Semáforos

```
int x = 0;  
Thread 1: x++;  
Thread 2: x++;
```

Qual o valor de x no final?

# Considerações

O problema da  
seção crítica

Semáforos

- ❑ Tipos básicos (e.g. int) são armazenados em posições de memória que são lidas e escritas de forma atômica
- ❑ Manipulação de valores depende da cópia deles para registradores, operações nesses registradores e cópia de volta para a memória
- ❑ Cada processo tem seu próprio conjunto de registradores (mudança de contexto)
- ❑ Resultados intermediários resultantes de expressões complexas são armazenados em registradores ou na memória particular de cada processo

# O problema da seção crítica

O problema da  
seção crítica

Semáforos

- $n$  processos/threads repetidamente executam uma seção de código crítica e uma seção de código não crítica

```
while (true) {  
    protocolo de entrada;  
    secao critica;  
    protocolo de saida;  
    secao nao critica  
}
```

- Considerando que um processo que entra na seção crítica, sairá alguma hora

O problema da  
seção crítica

---

▷ Semáforos

---

# Semáforos

# Para que semáforos?

O problema da  
seção crítica

Semáforos

- Surgem para evitar estados indesejáveis na execução de códigos com múltiplas threads
- (Nas estradas)
  - Mecanismos que sinalizam condições para garantir exclusão mútua de seções críticas da estrada.
- (Em programação concorrente)
  - Mecanismos básicos de sinalização usados para implementar exclusão mútua e sincronização

# O que é um semáforo?

O problema da  
seção crítica

Semáforos

- Um tipo especial de variável compartilhada proposto por Dijkstra
  - Assume valores inteiros não negativos
- Manipulado apenas por duas operações atômicas
  - V: usada para sinalizar a ocorrência de um evento
  - P: usada para atrasar um processo até que um evento ocorra

- Dijkstra era holandês
- V: holandês *verhogen* = incremento  
Incrementa o valor de um semáforo
- P: holandês *probeer te verlagen* = tentar reduzir  
Espera até o valor de um semáforo ser positivo e  
então decrementa o valor

# Em algoritmos

O problema da  
seção crítica

---

Semáforos

---

```
sem s; /* Semaforo s criado. Valor inicial = zero */  
  
sem lock = 1; /* Pode inicializar com qualquer valor  
              * nao negativo */  
  
sem forks[5] = ([5] 1); /* Vetor de semaforos criado  
                       * com todos = 1 */
```

# Em algoritmos

O problema da  
seção crítica

---

Semáforos

---

- semáforos são chamados de mutex quando o valor dos mesmos pode ser apenas 0 ou 1

# Em algoritmos

O problema da  
seção crítica

Semáforos

- As únicas formas de manipular semáforos são as operações P e V
- Cada operação recebe um semáforo como argumento e realiza **de forma atômica** algo com o efeito similar a:  
    P(s): while (s==0) skip; s = s - 1;  
    V(s): s = s + 1;
- V incrementa s de forma atômica
- P decrementa s de forma atômica mas apenas se s for positivo. Se não for, espera

# Resolvendo o problema da seção crítica com semáforos

O problema da  
seção crítica

Semáforos

- $n$  processos repetidamente executam a seção crítica
- a seção crítica requer acesso exclusivo para algum recurso compartilhado
- na seção não crítica são manipuladas variáveis locais

# Protocolo de acesso à seção crítica – com semáforos

O problema da  
seção crítica

Semáforos

```
sem mutex=1;

/* Cada thread roda: */
Thread() {
    while (true) {
        P(mutex);
        secao critica;
        V(mutex);
        secao nao critica;
    }
}
```

- Precisamos de uma forma para definir expressões como atômicas
  - Usaremos `< e >`
- O código: `<x++;>` é transformado em uma ação atômica

# Em algoritmos

O problema da  
seção crítica

Semáforos

```
P(s): <while (s==0) skip; s = s - 1;>  
V(s): <s = s + 1;>
```

- Considere  $s = 1$ 
  - Se dois processos tentarem rodar  $P(s)$  ao mesmo tempo
    - Apenas um vai conseguir
    - Se um processo tentar rodar  $P(s)$  e outro tentar rodar  $V(s)$ 
      - Os dois vão conseguir (em ordem imprevisível)
      - $s$  vai terminar igual a 1

# Regra importante ao implementar as operações

O problema da  
seção crítica

Semáforos

- Processos esperando em P são acordados na ordem em que eles executaram P
  - Uma quantidade adequada de execuções de V por outros processos garantirá que processos esperando em P eventualmente prosseguirão

# Em C, no Linux

O problema da  
seção crítica

Semáforos

- Semáforos genéricos: `semaphore.h`
- Tipo `sem_t`
- Funções `sem_init`, `sem_wait`, `sem_post` e `sem_destroy`

# Em C, no Linux

O problema da  
seção crítica

Semáforos

- Semáforos binários: `pthread.h`
- Obs.:** `apt-get install glibc-doc`
- Tipo `pthread_mutex_t`
- Funções `pthread_mutex_init`, `pthread_mutex_lock`,  
`pthread_mutex_unlock` e `pthread_mutex_destroy`