
MAC0438 – Programação Concorrente

Daniel Macêdo Batista

IME - USP, 21 de Maio de 2012

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

Introdução a monitores

Sintaxe e semântica

Sincronização com monitores

Disciplinas de sinalização

Monitores usando semáforos

▷ Introdução a monitores

Sintaxe e semântica

Sincronização com monitores

Disciplinas de sinalização

Monitores usando semáforos

Introdução a monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

☐ Semáforos

Facilitam a implementação de exclusão mútua

Facilitam a sinalização entre processos

Servem para implementar os `await`

☐ Entretanto...

Motivação

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Baixo nível
- ☐ Fácil para o programador se confundir (quantidade diferente de P's e V's)
- ☐ Difícil para depurar se o código for grande (variáveis globais)
- ☐ Difícil entender o propósito do código

Exclusão mútua usa P e V

Condição de sincronização também usa P e V!

O que são monitores?

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

□ Objetivos e definições

Visam eliminar as desvantagens de semáforos

Mecanismos de abstração de dados

Similares a classes

```
monitor nomedomonitor {  
    variaveis permanentes;  
    comandos de inicializacao;  
    demais procedimentos;  
}
```

O que são monitores?

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Encapsulam a representação de um objeto abstrato
- ☐ Fornecem um conjunto de operações
- ☐ A representação só pode ser manipulada por essas operações

Variáveis internas – Estado do objeto

Procedimentos – Alteram os valores das variáveis

- ☐ Processos só podem mudar o estado pelos procedimentos

Vantagens dos monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- Processos sempre vão chamar procedimentos dos monitores

O próprio procedimento internamente pode tratar da exclusão mútua

Fica mais próximo do objetivo do `<await>`

- Processos sempre tem que verificar as variáveis dos monitores

Monitores fornecem **variáveis de condição**

Similares a semáforos

Facilitam a implementação de condição de sincronização

Interação em programas usando monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- Processos ativos × Monitores passivos

Processos se comunicam modificando as variáveis
internas dos monitores

Modularização

- Um processo pode ignorar a implementação do monitor
- Os processos e os monitores podem ser implementados de forma independente
- Alto nível → Fácil de implementar e fácil de entender

O que vamos ver neste tópico

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Como usar monitores em algoritmos
- ☐ Como resolver diversos problemas de programação concorrente (alguns que já vimos bastante) usando monitores
- ☐ Como usar semáforos para implementar monitores e vice-versa
- ☐ Aplicação no EP3

Introdução a
monitores

▷ Sintaxe e
semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

Sintaxe e semântica

Interface e corpo

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Um monitor agrupa a representação e a implementação de um recurso compartilhado

- ☐ Interface

Apresenta as operações fornecidas pelo recurso compartilhado

- ☐ Corpo

Procedimentos que implementam as operações da interface

Variáveis que representam o estado do recurso

Em algoritmos

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

```
monitor nomedomonitor {  
    variaveis permanentes;  
    comandos de inicializacao;  
    demais procedimentos;  
}
```

☐ Variáveis permanentes

Como se fossem os atributos privados de uma classe

Compartilhadas por todos os procedimentos no
corpo do monitor

Mantêm os valores enquanto o monitor existir

Em algoritmos

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

```
monitor nomedomonitor {  
    variaveis permanentes;  
    comandos de inicializacao;  
    demais procedimentos;  
}
```

- Demais procedimentos

Como se fosse os métodos de uma classe

Implementam as operações visíveis

Só os nomes são divulgados para o exterior

- Comandos de inicialização

Como se fossem o construtor de uma classe

Executados quando o monitor é criado

Propriedades dos monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- Primeira consequência de ser uma instância de um tipo abstrato de dados:

Apenas os nomes dos procedimentos são visíveis fora do monitor

Chamadas aos procedimentos são as únicas formas de alterar o estado do monitor

```
call nomedomonitor.nomedoprocedimento(argumentos);
```

Propriedades dos monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- Segunda consequência de ser uma instância de um tipo abstrato de dados:

Comandos dentro dos monitores não podem acessar variáveis declaradas fora do monitor

Propriedades dos monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- Terceira consequência de ser uma instância de um tipo abstrato de dados:

Variáveis permanentes são inicializadas antes de qualquer procedimento ser chamado (comandos de inicialização)

Cuidado ao implementar monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- Monitores são compartilhados por processos que estejam executando de forma concorrente

O programador não sabe a priori como vai ser usado

É preciso ter certeza que as execuções arbitrárias dos procedimentos não vão levar o monitor a estados indesejados

Introdução a
monitores

Sintaxe e semântica

▷ Sincronização
com monitores

Disciplinas de
sinalização

Monitores usando
semáforos

Sincronização com monitores

Sincronização ideal

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

☐ Exclusão mútua

Implícita

Por exemplo não preciso me preocupar em proteger
 $x=x+1$ em dois processos

Facilita a leitura do código

Não há implementação de algoritmos para os
protocolos de entrada e saída da SC

Sincronização ideal

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

☐ Condição de sincronização

Explícita

Tem que especificar a condição a ser atendida

Poderia ser apenas a “tradução” da condição do `await` ou poderia ser algo mais baixo nível (facilidade × eficiência)

Exclusão mútua em monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Implícita
- ☐ No máximo uma instância de qualquer procedimento pode estar ativa por vez (Ativa = um processo está executando algum comando daquele procedimento)

Duas chamadas ao mesmo procedimento não podem estar ativas ao mesmo tempo

Duas chamadas a diferentes procedimentos não podem estar ativas ao mesmo tempo

Exclusão mútua em monitores

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ O programador que usa os monitores não precisa se preocupar
- ☐ A responsabilidade de garantir a exclusão mútua em monitores é da linguagem, biblioteca ou sistema operacional

Na prática, são usados por exemplo semáforos e desligamento de interrupções externas

Condição de sincronização

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

☐ Explícita

☐ Utiliza **variáveis de condição**

Usadas para atrasar um processo (O processo só pode continuar quando seu estado satisfizer alguma condição booleana)

Usadas para acordar um processo quando a condição torna-se verdadeira

```
cond nomedavariavel;
```


Condição de sincronização

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

```
cond nomedavariavel;
```

- ☐ O tipo `cond` é válido apenas dentro de monitores
- ☐ O valor de uma variável de condição é uma fila de processos atrasados

O valor da variável não é visível diretamente para o programador

Variáveis de condição

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Podem ser acessadas indiretamente

```
empty(nomedavariavel);
```

- ☐ Retorna verdade se a fila está vazia e falso caso contrário

Variáveis de condição

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

```
wait(nomedavariavel);
```

- ☐ Chamada que deve ser usada por um processo para bloqueá-lo em uma variável de condição
- ☐ O processo vai para o final da fila de `nomedavariavel`
- ☐ **Enquanto estiver esperando, o acesso exclusivo ao monitor é liberado para outros processos**

Se não fosse assim os monitores causariam deadlock

Variáveis de condição

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

```
signal(nomedavariavel);
```

- Acorda outros processos que estejam bloqueados na fila de `nomedavariavel`

Se a fila está vazia, nada acontece

Se a fila tem algum processo, aquele na primeira posição da fila é acordado

`wait` e `signal` fornecem uma disciplina de sinalização FIFO

É possível adicionar prioridades (veremos nas próximas aulas)

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

▷ Disciplinas de
sinalização

Monitores usando
semáforos

Disciplinas de sinalização

O que acontece quando o `signal` é executado?

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Há 1 processo executando (o processo que chamou o `signal`)
- ☐ Esse processo é o único com permissão para acessar o monitor (por definição)
- ☐ Supondo que haja algum processo na fila da variável do `signal`, o que acontece quando o `signal` é executado?
- ☐ **DILEMA!**

O que acontece quando o signal é executado?

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Após o signal vai haver dois processos que podem executar

O processo acordado

O processo que executou signal

- ☐ Pela definição de monitor apenas um processo pode executar

- ☐ Qual deve assumir o “controle” do monitor?

O que acontece quando o signal é executado?

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ O processo que executará após o signal é definido pela disciplina de sinalização

- ☐ **Sinaliza e continua (Signal and Continue)**

O processo que sinalizou continua e o processo acordado executará alguma hora no futuro

- ☐ **Sinaliza e espera (Signal and wait)**

O processo acordado executa e o processo que sinalizou espera para executar alguma hora no futuro

Sinaliza e continua

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Disciplina não-preemptiva
- ☐ O `signal` serve como uma dica de que o processo acordado vai poder executar
- ☐ O processo acordado sai da fila da variável de condição mas entra na fila para acesso ao monitor

Sinaliza e espera

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ Disciplina preemptiva
- ☐ O processo que chamou o `signal` entra na fila para acesso ao monitor

Sinaliza e espera

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- ☐ A disciplina Sinaliza e espera pode ser modificada para que o processo que chamou o `signal` fique no início da fila para acesso ao monitor
- ☐ **Sinaliza e espera com urgência (Signal and Urgent Wait)**

Resumo da sincronização em monitores

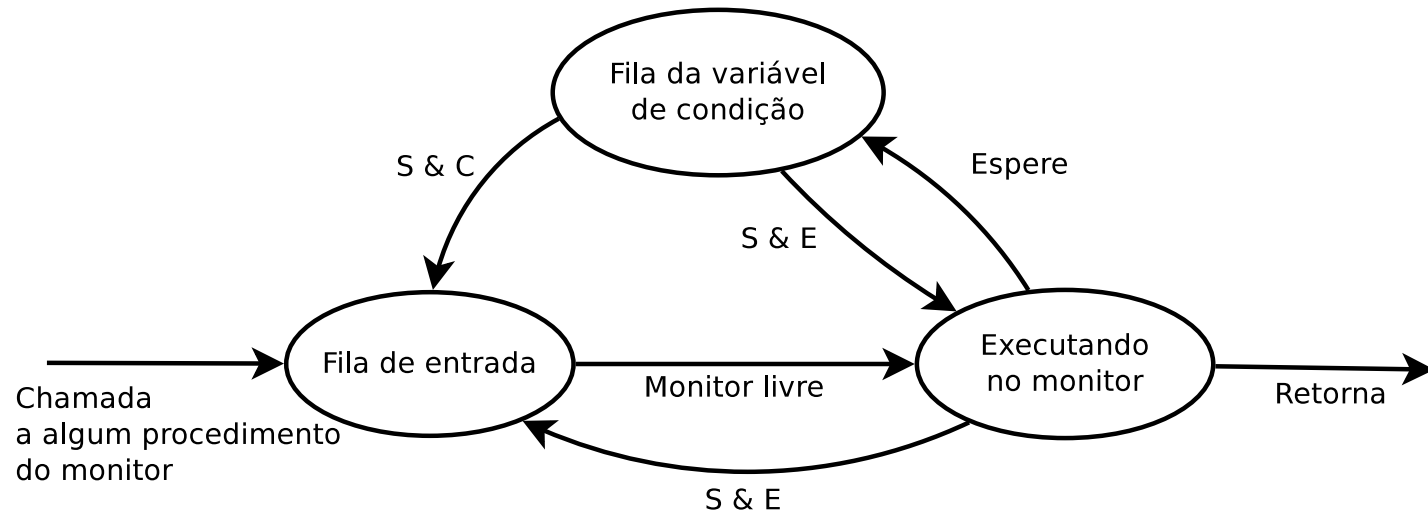
Introdução a monitores

Sintaxe e semântica

Sincronização com monitores

Disciplinas de sinalização

Monitores usando semáforos



Resumo

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

```
monitor nomedomonitor {  
    variaveis permanentes;  
    comandos de inicializacao;  
    demais procedimentos;  
}
```

- ❑ Variáveis permanentes → Atributos privados de uma classe
- ❑ Comandos de inicialização → Construtor de uma classe
- ❑ Demais procedimentos → Métodos de uma classe

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- Única forma de modificar as variáveis é através dos procedimentos

```
call nomedomonitor.nomedoprocedimento(argumentos);
```

☐ Exclusão mútua

Implícita

No máximo uma instância de qualquer procedimento pode estar ativa por vez (Ativa = um processo está executando algum comando daquele procedimento)

☐ Condição de sincronização

Explícita

Utiliza **variáveis de condição**

```
cond nomedavariavel;
```

Variáveis de condição

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- Manipuladas dentro do monitor por diversas funções

```
empty(nomedavariavel);
```

```
wait(nomedavariavel);
```

```
signal(nomedavariavel);
```


Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

▷ Monitores usando
semáforos

Monitores usando semáforos

Como implementar um semáforo usando monitor?

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

- O monitor terá dois procedimentos:

Psem: decrementa o semáforo se ele for positivo.
Enquanto não for, espera.

Vsem: incrementa o semáforo

- O monitor terá uma variável de condição

A espera do procedimento Psem vai usar o `wait` em
uma variável de condição

O processo na fila da variável de condição será
acordado por um `signal` do procedimento Vsem

- Obs.: monitores também podem ser implementados usando
semáforos

Um semáforo implementado com monitor

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

```
monitor Semaforo {  
    int s = 0; /* Valor do semaforo (s >= 0) */  
    cond pos; /* Sinaliza quando s > 0 */  
  
    procedure Psem() {  
        if (s == 0) wait(pos);  
        s = s-1;  
    }  
  
    procedure Vsem() {  
        s = s+1;  
        signal(pos);  
    }  
}
```

Verificando o semáforo

Introdução a
monitores

Sintaxe e semântica

Sincronização com
monitores

Disciplinas de
sinalização

Monitores usando
semáforos

☐ Variáveis permanentes e inicialização

```
int s = 0; /* Valor do semaforo (s >= 0) */  
cond pos; /* Sinaliza quando s > 0 */
```

- ☐ O semáforo vai ser inicializado com 0 (poderia ser qualquer valor)
- ☐ A variável de condição pos vai gerenciar uma fila de processos esperando o valor do semáforo ser positivo