

© 2005-2009 Volnys Bernal 1

Threads

Volnys Borges Bernal
volnys@lsi.usp.br
http://www.lsi.usp.br/~volnys



© 2005-2009 Volnys Bernal 2

Agenda

- ❑ Processo
- ❑ Threads
 - ❖ Interface de threads
 - ❖ Uso de threads
- ❑ Interfaces de threads
- ❑ Uso de threads
- ❑ Threads de usuário x threads de núcleo
 - ❖ Threads de usuário
 - ❖ Threads de núcleo

© 2005-2009 Volnys Bernal 3

Processo



© 2005-2009 Volnys Bernal 4

Processo



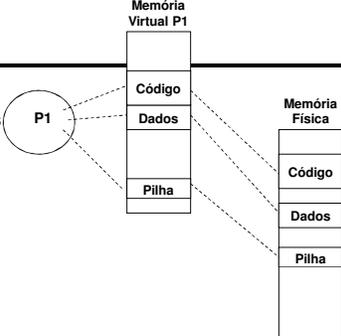
© 2005-2009 Volnys Bernal 5

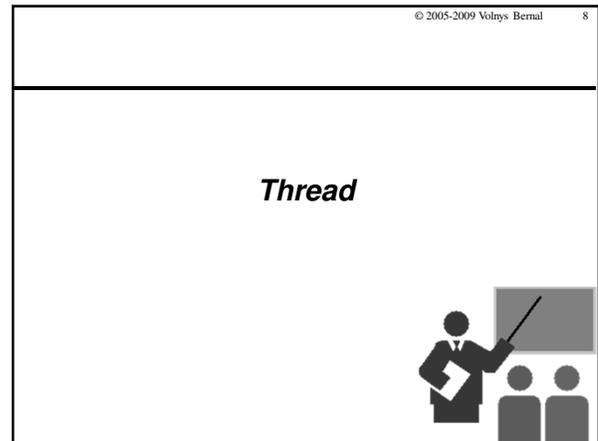
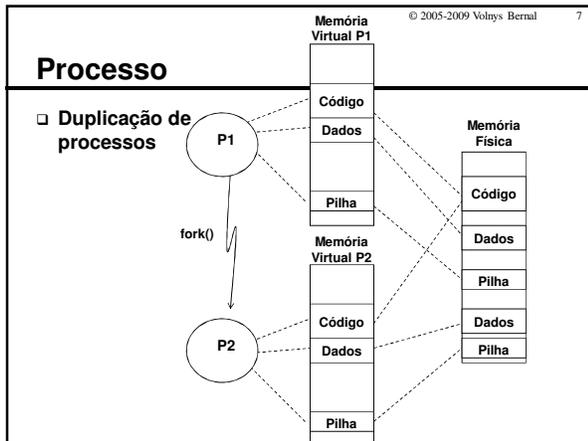
Processo

- ❑ Um processo tradicional:
 - ❖ Um único thread (linha de execução)
 - ❖ Que executa sobre um espaço de endereçamento próprio
- ❑ Espaço de endereçamento
 - ❖ Área de código
 - ❖ Área de dados
 - ❖ Área da pilha de execução
- ❑ Linha de execução
 - ❖ Sequência de instruções executadas
 - ❖ Controlada pelo pelo registrador PC (*Program Counter*)
 - ❖ Em decorrência do estado do contexto
 - áreas de memória, valores de registradores, etc

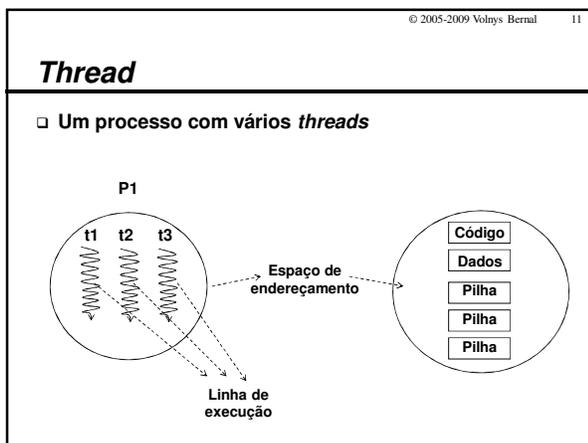
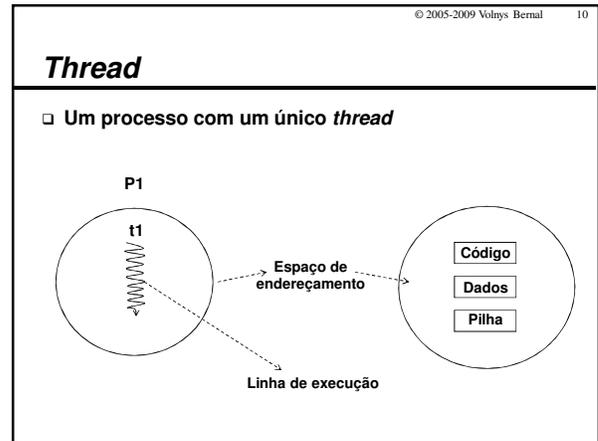
© 2005-2009 Volnys Bernal 6

Processo

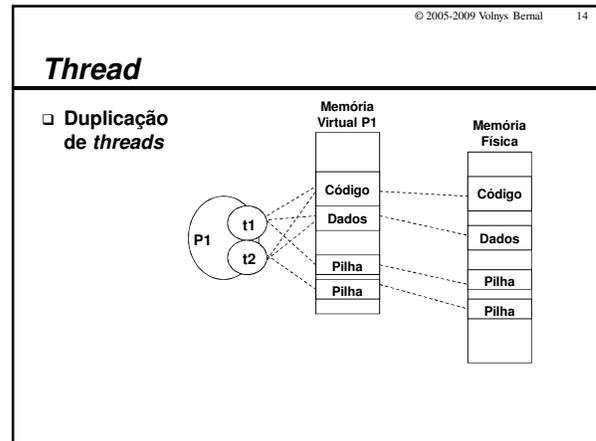
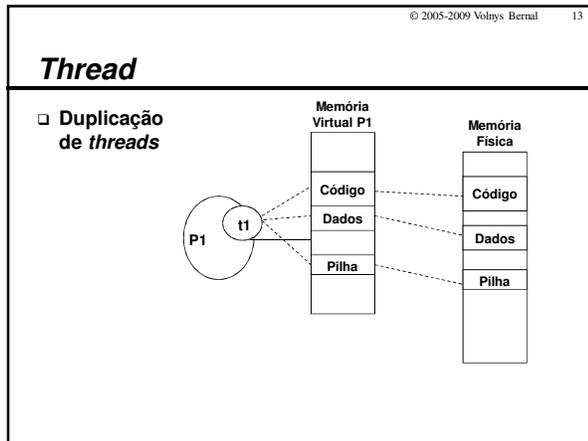
- ❑ Duplicação de processos
 



- © 2005-2009 Volnys Bernal 9
- ### Thread
- ❑ Thread = Linha de execução
 - ❑ O que é?
 - ❖ Componente do processo relacionado ao fluxo de execução
 - ❖ Thread permite separar os componentes relacionados ao fluxo de execução dos outros componentes de um processo.
 - ❑ Em sistemas operacionais modernos:
 - ❖ Um processo pode ser composto por um ou mais threads
 - ❖ Alguns recursos ficam associados ao processo, outros a cada thread
 - ❑ Os recursos dependentes diretamente da execução ficam associados ao thread:
 - ❖ Informações de controle do thread
 - Identificação do thread
 - Área para salvamento de registradores
 - Estado do thread
 - ❖ Registradores de CPU
 - ❖ Área da pilha de execução



- © 2005-2009 Volnys Bernal 12
- ### Thread
- ❑ Itens associados ao processo
 - ❖ Identificação do processo
 - ❖ Espaço de endereçamento
 - ❖ Threads
 - ❖ Arquivos abertos
 - ❖ Processos filhos
 - ❖ Alarmes
 - ❖ Sinais e tratadores de sinais
 - ❖ Informações de contabilidade
 - ❑ Itens associados ao thread
 - ❖ Identificação do thread
 - ❖ Registradores
 - ❖ Área da pilha de execução
 - ❖ Estado



© 2005-2009 Volnys Bernal 15

Thread

❑ O estado de um processo é, na verdade, o estado do *thread* principal

❑ Estados de um thread:

- ❖ Pronto
- ❖ Executando
- ❖ Bloqueado
- ❖ Terminado

❑ Paralelismo x concorrência

- ❖ Hardware monoprocessador
 - Concorrência entre *threads*
- ❖ Hardware multiprocessador
 - Concorrência e paralelismo entre *threads*

© 2005-2009 Volnys Bernal 16

Interfaces de threads

© 2005-2009 Volnys Bernal 17

Interfaces de Threads

❑ Existem várias interfaces para manipulação de threads

- ❖ Pthreads (IEEE POSIX 1003.1c)
- ❖ Process Share Group (IRIX)
- ❖ User Level Threads (Solaris)
- ❖ Light Weight Process (Solaris)
- ❖ Cthreads (OSF/Mach)

© 2005-2009 Volnys Bernal 18

Interfaces de Threads

❑ Interface pthreads

❖ Chamadas básicas para controles de threads:

- Pthread_create()
 - Cria um novo thread para o mesmo processo
- Pthread_exit()
 - Termina o thread
- Pthread_join()
 - Aguarda um thread terminar
- Pthread_yield()
 - Libera o processador para outro thread

© 2005-2009 Volnys Bernal 19

Uso de threads



© 2005-2009 Volnys Bernal 20

Uso de threads

- Principais usos
 - ❖ Processamento paralelo (sistemas multiprocessadores)
 - ❖ Sistemas interativos ou que controlam “interfaces” simultâneas
 - Interface com usuário
 - Interface de rede (transmissão e recepção)
 - Interface com dispositivos

© 2005-2009 Volnys Bernal 21

Uso de threads

- Exemplo de servidor WEB monothread:

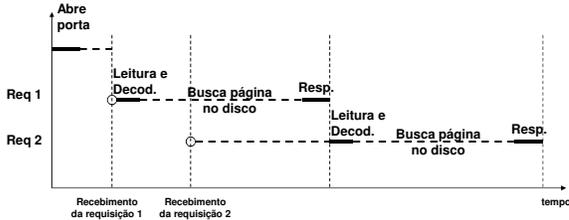
```

ServidorWEB ()
{
  Abre porta TCP/80
  While (TRUE)
    Aguarda conexão
    Leitura da mensagem HTTP
    Decodifica requisição HTTP
    Busca página no disco
    Responde requisição
}
    
```

© 2005-2009 Volnys Bernal 22

Uso de threads

- Exemplo de servidor WEB monothread:



© 2005-2009 Volnys Bernal 23

Uso de threads

- Exemplo de servidor WEB multithread:

```

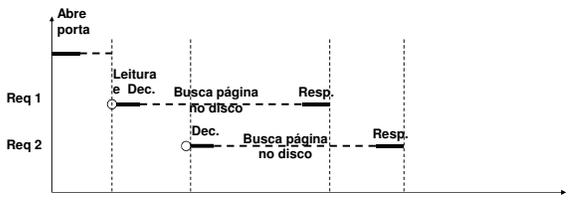
ServidorWEB ()
{
  Abre porta TCP/80
  While (TRUE)
    Aguarda conexão
    Dispara thread operário
}

ThreadOperário ()
{
  Leitura da mensagem HTTP
  Decodifica requisição HTTP
  Busca página no disco
  Responde requisição
}
    
```

© 2005-2009 Volnys Bernal 24

Uso de threads

- Exemplo de servidor WEB multithread:



© 2005-2009 Volnys Bernal 25

Threads de usuário X Threads de núcleo



© 2005-2009 Volnys Bernal 26

Threads de usuário x threads de núcleo

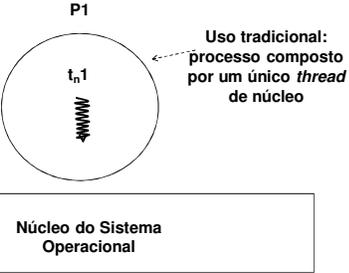
❑ **Existem 2 formas de implementação de threads:**

- ❖ **Threads de usuário**
 - A abstração de threads é criada por um conjunto de rotinas de biblioteca utilizada pelo próprio processo
 - Vantagem: facilidade de implementação (fora do núcleo)
 - Desvantagem: controle do momento da troca de contexto; bloqueio de todos os threads em chamadas ao sistema bloqueantes
- ❖ **Threads de núcleo**
 - A abstração de threads é criada pelo núcleo do sistema operacional
 - Vantagem: precisão do gerenciamento dos threads
 - Desvantagem: sobrecarga de processamento de gerenciamento

© 2005-2009 Volnys Bernal 27

Threads de usuário x threads de núcleo

❑ **Threads de usuário**

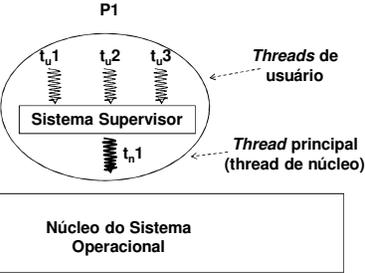


Núcleo do Sistema Operacional

© 2005-2009 Volnys Bernal 28

Threads de usuário x threads de núcleo

❑ **Threads de usuário**



Núcleo do Sistema Operacional

© 2005-2009 Volnys Bernal 29

Threads de usuário x threads de núcleo

❑ **Threads de usuário**

- ❖ **Vantagens**
 - Pode ser implementado por um sistema operacional que não suporte a abstração de threads
 - (atualmente a maior parte dos sistemas operacionais fornecem a abstração de threads)
 - Troca de contexto entre os threads de usuário não envolve a passagem de modo usuário para modo supervisor (chamada ao sistema), sendo muito mais rápida
 - Possibilita customizar o algoritmo de escalonamento entre os threads de usuário do próprio processo

© 2005-2009 Volnys Bernal 30

Threads de usuário x threads de núcleo

❑ **Threads de usuário**

- ❖ **Desvantagens**
 - Chamadas ao sistema bloqueantes
 - Thread de usuário não é interrompido pelo escalonador
 - Sistema supervisor não atende interrupções do temporizador
 - Cada thread de usuário deve ceder a CPU (thread_yield), de tempos em tempos, para os outros threads de usuário
 - Poderia estar requisitando um alarme do S.O periódico
 - → Problema: sobrecarga de processamento

