unipampa
Universidade Federal do Pampa

Sistemas Operacionais

10 setembro - 16 setembro

Agenda

- Apresentação
- Metodologia
- Processos
  - O que é um processo?
  - Ciclo de vida de um processo

Apresentação

- **Bacharel em Ciências da Computação** | 2018 - 2022
  - Universidade Federal do Pampa

- **Mestrado em Engenharia de Software** | 2024
  - Universidade Federal do Pampa

Apresentação

- **Desenvolvedor Back-End Pleno** | Desde Junho de 2024
  - Contabilizei

- **Desenvolvedor de Sistemas** | Março de 2022 - Junho de 2024
  - Universidade de Santa Cruz do Sul

Apresentação

Linkedin

https://www.linkedin.com/in/glenerpizzolato

GitHub

https://github.com/glener10

Apresentação

- Nome
- Idade
- Onde mora
- Curso / Semestre
- Alguma curiosidade

Metodologia

- Momento teórico durante a aula

Metodologia

- Momento teórico durante a aula
- Momento prático
  - **A atividade será simples para poder fazer no mesmo momento da aula**
    - Ficarei online durante o momento da aula enquanto tiver aluno para tirar dúvidas
  - Entrega até o domingo **22/09**

Metodologia

- Momento teórico durante a aula
- Momento prático
  - **A atividade será simples para poder fazer no mesmo momento da aula**
    - Ficarei online durante o momento da aula enquanto tiver aluno para tirar dúvidas
  - Entrega até o domingo **22/09**
- Não deixe de fazer a entrega, qualquer dúvida entre em contato 👇
  - glenerpizzolato.aluno@unipampa.edu.br
    - Preferência usar o Chat
    - OBS: Só verei a noite

Metodologia

- Momento teórico durante a aula
- Momento prático
  - **A atividade será simples para poder fazer no mesmo momento da aula**
    - Ficarei online durante o momento da aula enquanto tiver aluno para tirar dúvidas
  - Entrega até o domingo **22/09**
- Não deixe de fazer a entrega, qualquer dúvida entre em contato 👇
  - glenerpizzolato.aluno@unipampa.edu.br
    - Preferência usar o Chat
    - OBS: Só verei a noite
- Qualquer dúvida pode interromper a aula

Processos

- Process Control Block (PCB)
- Relacionamento/Comunicação entre processos
- Modos de execução
  - Privilegiado
  - Usuário
- Gerenciamento de memória

Processos

- Um **processo** é:
  - Uma abstração que representa um programa ***em execução***;
  - Uma entidade ***dinâmica***: seu estado se altera conforme for executando.
  - Armazenado na memória: composto por programa (código), dados e contexto (valores)

Processos

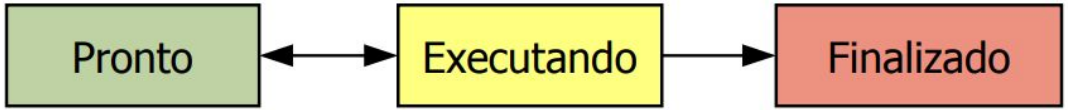
- Quando se **cria** um processo
  - Chama-se o **construtor**
    - Ele aloca um novo PCB preenchendo-o com os dados iniciais
    - Inicializa (se for o caso) as listas de processos
- Durante o **ciclo de vida**
  - O PCB passa de uma lista para um outra (pronto, bloqueado,...);
  - Executam-se as instruções apontadas (PC)
  - Altera os campos do PCB conforme necessário (idem para o uso dos recursos).
- Quando o processo **finaliza**
  - Chama-se o **destrutor**;
  - Libera-se a memória assim como os recursos usados;

Processos

- Processos nascem
  - No momento de sua criação (via chamada de sistema: fork, spawn…)
- Processos vivem
  - Alternam entre: executar na CPU e liberar a CPU (E/S)
    - Via ***chamada de sistema***, ***interrupção***, ou por causa de um ***evento***.
  - Os processos podem ser:
    - Programas dos usuários
    - Programas do sistema (*daemons*)
- Processos morrem
  - Ou porque terminaram sua execução
  - Ou porque um outro processo os finalizou:
    - Erro, acesso não-autorizado, falha.

Processos

- Ao ser criado, o processo está *pronto* para usar a CPU.
  - O que acontece se a CPU não está disponível?
  - O que acontece se vários processos estão sendo criados ao mesmo tempo?
  - É necessário manter uma lista de processos prontos!
- Ao executar, o processo pode requerer de E/S:
  - O que acontece se o recurso de E/S está ocupado?
  - É preciso manter uma fila de processos *bloqueados*
- Após ter executado, o processo passa a estar *encerrado*.



Processos - Ciclo de vida

- Fila de jobs
  - Conjunto de todos os processos do sistema
- Fila de prontos
  - Conjunto de todos os processos residindo na memória principal, prontos e esperando para executar
- Filas de dispositivos
  - Conjunto de processos esperando por um dispositivo de E/S
- Processos migram entre diversas filas

Processos - Ciclo de vida

- **Em resumo:**
  - Existem 5 estados possíveis para um processo:
    - Criado
    - Pronto
    - Executando
    - Bloqueado
    - Encerrado
  - Durante o ciclo de vida de um processo ocorrem transições entre esses 5 estados

Processos - Ciclo de vida

- Enquanto executam, os processos apresentam 2 tipos básicos de comportamento:
  - Ou usam a CPU
    - Cálculo, operações sobre a memória/registradores
  - Ou fazem Entrada/saída – liberam a CPU
    - Escrita na tela, entrada teclado, som, vídeo, rede, disco…
- Logo, têm-se processos ***CPU-bound*** vs. ***I/O-bound***
- **Caso ideal:**
  - Mesclar os dois tipos de processos através de escalonamento visando obter um melhor aproveitamento da CPU
- **Na prática:**
  - É difícil distinguir processos *CPU-bound* de *I/O-bound* e vice-versa

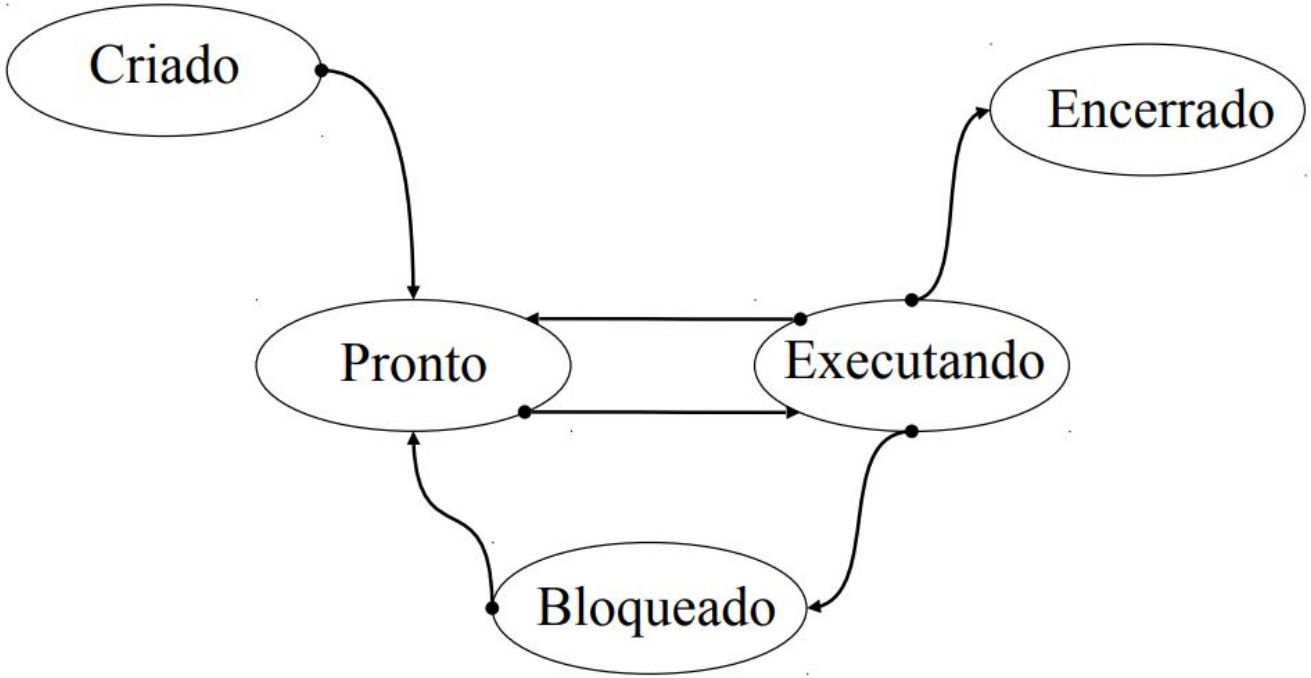
Processos - Ciclo de vida

- **Pronto -> executando**
  - Algoritmo de escalonamento

Próxima aula

- **Executando -> pronto**
  - Interrupção de tempo
  - Interrupção devida ao escalonador
  - Decisão espontânea (yield)
- **Executando -> bloqueado**
  - E/S
  - Sincronização
- **Bloqueado -> pronto**
  - Interrupção
- **Executando -> encerrado**
  - Interrupção (CTRL-C)
  - Término normal
- **Bloqueado, pronto -> encerrado**
  - interrupção

Processos - Ciclo de vida



(Silberschatz, 4.1.2)

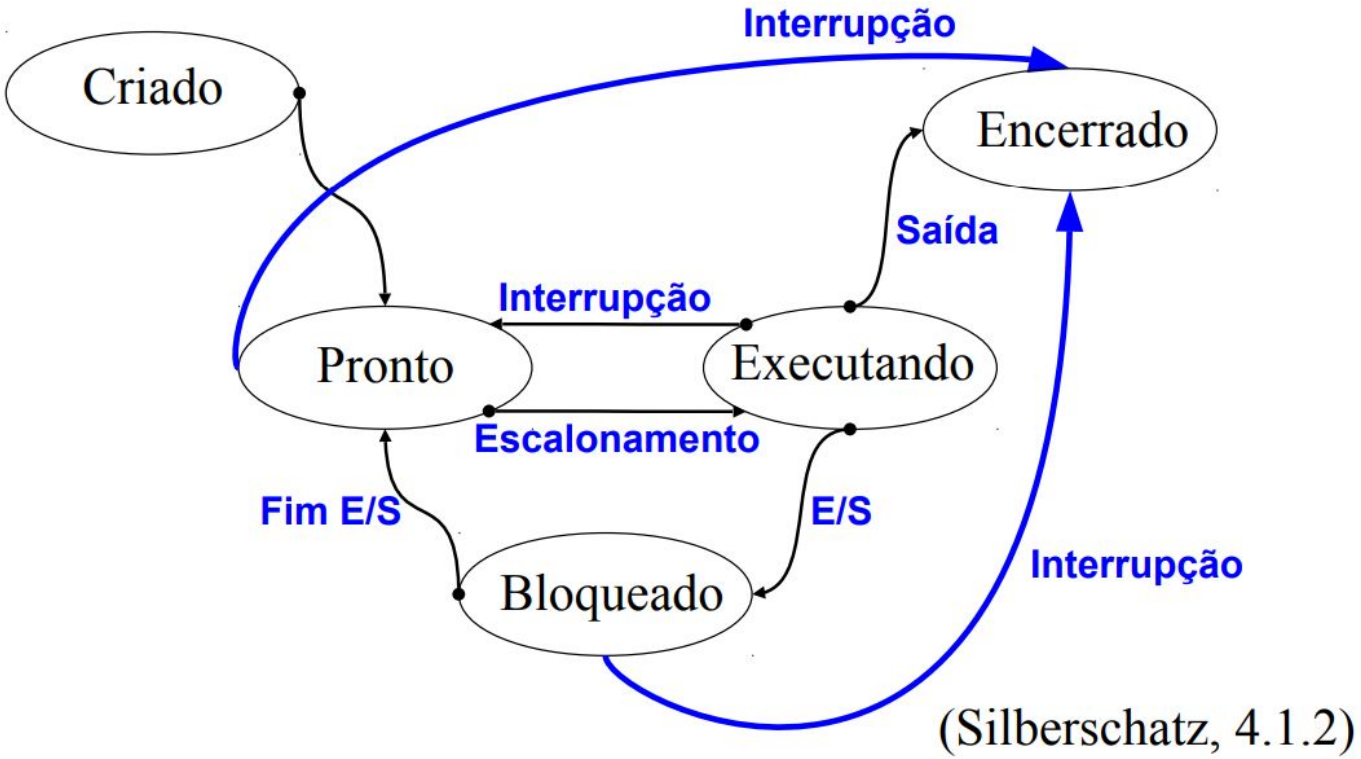
Processos - Ciclo de vida

- Erros e eventos são detectados por hardware
  - Exemplos de evento: inserir um pendrive na porta USB, escrever um bloco em disco, receber um pacote pela rede, escrever numa área proíbida…
- O HW emite uma **interrupção**
  - Uma interrupção transfere o controle para a **rotina de atendimento da interrupção**, por meio do vetor de interrupções, que contém o endereço de todas as rotinas de serviço.
- A instrução em execução na CPU precisa ser salva para que se possa retornar a ela após o tratamento da interrupção

Processos - Ciclo de vida

- O Sistema Operacional trata uma interrupção:
  - identificando-a (por um número);
  - verificando sua prioridade;
  - achando no vetor de interrupções qual procedimento deve ser realizado (*handler*).
- O que acontece se houver uma interrupção durante o tratamento de uma interrupção?
  - Compara-se as prioridades
  - Desabilita-se as interrupções para evitar perdas.

Processos - Ciclo de vida



(Silberschatz, 4.1.2)

Processos - Mais um Estado?

- Dois problemas principais para gerenciar os recursos:
  - A CPU é muito mais rápida do que a memória;
  - A memória é de tamanho finito.

É essencial poupar **memória**

- Logo:
  - Processos bloqueados que estejam na memória podem ser transferidos para o disco (*swap*) até que sua E/S acabe
- Processos prontos podem também ser transferidos para o disco.
- Chega-se a mais dois estados:
  - **Bloqueado, suspenso.**
  - **Pronto, suspenso.**

Processos - Prática no terminal

- Criação de processo: *fork()*
  - Cria um novo processo
  - Igual ao pai (clone)
  - No processo pai, fork() retorna o pid do filho;
  - No processo filho, fork() retorna 0.

- Mudar o segmento de código: *exec()*
  - Executa o binário apontado em argumento.
  - Em geral, chamado logo após o fork() (“fork-exec”)

- Recuperar o identificador: *getpid()*
  - Retorna um int, que identifica o processo.

- Terminar o processo: *kill()*
  - Manda um sinal (e.g. TERM) para o processo cujo pid é dado em argumento.

Processos - Trabalho Prático

- Leitura da atividade
- Demonstração de implementação

Exemplo

https://github.com/glener10/aulas-SO

Obrigado pela atenção!

Perguntas?

- [1] Glener L. Pizzolato
  - glenerpizzolato.aluno@unipampa.edu.br

Algumas referências e slides foram retirados de conteúdos do prof Dr. Claudio Schepke

unipampa
Universidade Federal do Pampa