Atividades Práticas no Ensino Introdutório de Sistemas Operacionais

Cassio P. de Campos Nicolas Kassalias

Faculdade de Computação e Informática – Universidade Mackenzie

17 de julho de 2006

Agenda

Agenda

Ensino de Sistemas Operacionais

Gerenciamento de Recursos? :-)

Objetivo

O objetivo é apresentar nossa experiência com atividades práticas no ensino de Sistemas Operacionais.

Ambiente

Simuladores: NachOS, MOSS, OSS, SOsim, SOS, POPSS, ...
Sistemas Operacionais: Minix, Linux, ...

Agenda

Sistema Operacional Minix

- Desenvolvido com foco no ensino de Sistemas Operacionais.
- Sistema Unix-like antecessor do Linux.
- Escrito essencialmente na linguagem de programação C.
- Na versão 2, núcleo com cerca de vinte mil linhas de código, simples e bem documentado.

Minix 3

Em 2005 foi anunciada uma nova versão do Minix. Embora ainda sirva como exemplo didático, a nova versão foi reescrita para ser mais difundida. Com as modificações, seu núcleo tem menos de quatro mil linhas de código.

Bochs

- Emulador de PC de código aberto altamente portável.
- Inclui a emulação da CPU, dispositivos e BIOS de um PC.
- Capaz de executar diversos sistemas operacionais, como Linux, DOS, Windows e Minix.
- Grande suporte aos dispositivos de um PC.
- Desvantagem está na eficiência: executa os códigos de forma semelhante a um processador de *scripts*, ou seja, faz a tradução em tempo de execução.
- Disponível para os sistemas Linux e Windows, podendo ser rapidamente instalado pelos alunos em qualquer computador pessoal.

QEMU

- Emulador genérico e de código aberto que tem excelente desempenho.
- Emula um sistema computacional completo, incluindo o processador e periféricos.
- Faz tradução dinâmica de código.

Tradução dinâmica de código

A idéia básica é quebrar cada instrução em algumas novas instruções mais simples. Cada instrução simples está implementada por um pequeno trecho de código. Uma ferramenta de compilação recebe um arquivo objeto e gera um código dinâmico pela concatenação das instruções simples da máquina hospedeira.

Vantagens e desvantagens

- Diminuem o trabalho de instalação nos laboratórios (que normalmente são compartilhados entre dezenas de disciplinas com propósitos diferentes).
- Aumentam a estabilidade e facilidade de manipulação do sistema, já que estamos fazendo constantes alterações no núcleo.
- Aumentam a dificuldade na manipulação dos arquivos devido ao transporte de arquivos internos (emulador) e externos (sistema operacional hospedeiro).

Agenda

Introdução

Acoplamento

Aulas teóricas e atividades práticas devem estar acopladas, obrigando o aluno a estar atento às explicações.

Relatórios

Relatórios detalhados com explicações sobre os testes realizados e código alterado são importantes na avaliação.

Principal objetivo

O foco das atividades está na experiência prática de trabalhar com os conceitos abordados na parte teórica. Não há pretensão de novas idéias ou melhorias na qualidade da implementação.

Objetivo

Atividade inicial simples, com objetivo de ambientar os alunos.

- Primeira alteração do núcleo do sistema: toda vez que uma tecla específica é pressionada, deve-se imprimir na tela uma mensagem simples.
- Segunda alteração: toda vez que a tecla é pressionada, imprimimos o número de trocas de contexto.

Significado de trocas de contexto

Inicialmente não indicamos o significado do número que está sendo calculado, e o aluno apenas percebe que estamos contando o número de vezes que uma dada função é executada. Após testes e o andamento da parte teórica da disciplina, isso é clarificado.

Objetivo

Trabalhar com chamadas de sistema, idéias de processos executando "simultaneamente" e a tabela de processos.

- Chamadas de sistema *fork*, *wait*, *sleep*.
- Desenvolvimento de código que inicializa diversos filhos simultâneos.
- Verificação e alteração das características da tabela de processos.

fork bomb

Alunos sobrecarreguem o sistema e analisam as conseqüências (incluindo seu travamento). Então solicitamos a alteração do núcleo, incluindo uma limitação da quantidade de processos que cada usuário pode criar.

Objetivo

Escalonamento é parte crítica do sistema. Perceber que pequenas alterações e maneiras diferentes de tratar o problema tendem a mudar significativamente o desempenho global.

- Mudança de parâmetro em tempo de execução no algoritmo de escalonamento. Interação pelo teclado aumenta ou diminui o *quantum*.
- Remoção das linhas de código que fazem a troca de processos na fila de execução.
- Alteração do algoritmo de escalonamento (Round-Robin para Garantido) e testes sobre eficiência.

Atividades anteriores

Para testar e comparar os algoritmos de escalonamento, podemos utilizar as atividades anteriores.

Objetivo

Trabalhar com programação concorrente e comunicação entre processos. Fixar idéias de regiões críticas e soluções para o problema da exclusão mútua.

- Apresentação prática dos problemas envolvendo variáveis de bloqueio, alternação estrita, solução de Peterson.
- Uso de semáforos para problemas clássicos, como produtor-consumidor e filósofos.

Dificuldade dos alunos

Notamos com alunos inexperientes que é necessária uma ajuda especial. Sugerimos, nos casos onde o tempo é limitado, disponibilizar (de forma total ou parcial) os códigos-fonte, e deixá-los responsáveis apenas pela condução de testes e pequenas alterações.

Objetivo

Conhecer o sistema de gerenciamento de memória.

- Depuração (através de informações na tela) sobre cada alocação ou liberação de memória.
- Criação de código que se utilize de memória de forma aleatória. Utilização desse código para testes.

memory bomb

A criação e teste de código que aloca memória "infinitamente" mostra um problema a ser tratado.

Alteração do gerenciamento de memória

Minix não tem memória virtual ou *swapping*. A inclusão de um esquema de memória virtual ou *swapping* é interessante, porém usualmente maior do que o tempo disponível em uma disciplina introdutória.

Objetivo

Conhecer um sistema de arquivos. Mostrar uma alteração prática que poderia ser incorporada ao dia-a-dia.

- Inclusão de um procedimento de lixeira no sistema de arquivos. Arquivos são movidos para um diretório especial no lugar de serem diretamente removidos.
- Cuidados com a remoção de diretórios e arquivos com nomes repetidos.

Partições em ambientes Unix

As partições são "alocadas" dentro de uma única estrutura de diretórios (tendem a ser transparentes ao usuário). O sistema operacional e o sistema de arquivos devem tratar alguns casos, como a diferença em mover/copiar arquivos na mesma partição ou em partições diferentes.

Agenda

Disciplina de Sistemas Operacionais

- Sete oferecimentos de Sistemas Operacionais para o bacharelado em Ciência da Computação - Mackenzie.
- A disciplina é semestral e tem carga horária de quatro horas semanais, sendo duas horas teóricas em sala e duas horas em laboratórios de informática.
- Atividade prática proposta a cada duas semanas.
- Com as atividades, obtivemos aumento do interesse dos alunos e conseqüente diminuição no número de reprovações.
- Aulas práticas podem ser presenciais ou remotas.

Sistemas Operacionais no nível introdutório

Atividades curtas

Seqüência de atividades curtas é mais produtiva que um projeto mais elaborado e extenso.

Atividades extensas

Projeto extenso tem melhor aplicação quando os alunos já têm algum conhecimento prévio sobre o assunto.

Desafios reais

Motivação dos alunos em tratar situações relacionadas à realidade do dia-a-dia.

- Aulas práticas em laboratório colaboram no ensino de Sistemas Operacionais.
- Aulas presenciais podem ser substituídas pela utilização de algum ambiente de ensino não-presencial.
- Atividades curtas em maior quantidade mostraram melhores resultados em motivação e aprendizado que atividades longas em menor número.
- Maior motivação quando sugerimos:
  - Alterações reais em sistemas operacionais utilizados em larga escala.
  - Construção de programas capazes de desafiar a estabilidade e segurança do sistema.

Obrigado

Dúvidas, Comentários, Sugestões

cassio@ime.usp.br