Fundamentos de Sistemas Operacionais
Atividade Extra 02

Prof. Tiago Alves

Chamadas de Sistemas: System Calls

Introdução

A disciplina de Fundamentos de Sistemas Operacionais trata de diversos tópicos desses sistemas que provêem uma forma intuitiva de se utilizar as funcionalidades de computadores digitais sem que seja necessário ao usuário ou programador ter profundo conhecimento das interações entre os diferentes *hardwares* que compõem um computador.

Para construir ou adicionar funcionalidades a esses sistemas computacionais, é necessário conhecimento de linguagens de programação e ferramentas de desenvolvimento. Em nosso curso, o domínio da linguagem C é um pré-requisito para o devido acompanhamento das atividades da disciplina.

Objetivos

1) Exercitar conceitos da linguagem de programação C, especialmente aqueles referentes à programação de sistemas operacionais.
2) Criar uma nova chamada de sistema para sistema operacional *nix.

Referências Teóricas

Mitchell, Mark, Jeffrey Oldham, and Alex Samuel. *Advanced linux programming*. New Riders, 2001.

Wolfgang Mauerer, *Professional Linux kernel architecture*, Wrox_Wiley Pub, 2008.

Kernel do Linux: kernel.org

Material Necessário

- Computador com sistema operacional programável
- Ferramentas de desenvolvimento GNU/Linux ou similares: compilador GCC, depurador, editor de texto.

Roteiro

1) Revisão de técnicas e ferramentas de desenvolvimento de módulos de Kernel Linux.

Colete o material acompanhante do roteiro do trabalho a partir do Moodle da disciplina e estude os

princípios e técnicas de desenvolvimento de implementação de chamadas de sistema no operacional Linux.

2) Realizar as implementações solicitadas no questionário do trabalho.

Implementações e Questões para Estudo

1) Adicione uma nova chamada de sistema ao Kernel do Linux de forma a contemplar os seguintes requisitos:
   - Assinatura: int getproctimes(int pid, struct proctimes *pt);
     - Primeiro argumento: pid do processo ou processo atual (se pid==-1)
     - Segundo argumento: passado por referência e usado pelo kernel para retornar as informações necessárias ao espaço de usuário.
     - Retorna EINVAL em caso de erro ou 0, se sucesso.
   - Cada chamada a getproctimes deverá ser executada no kernel e imprimir uma mensagem:
     - Use a primitiva **printk** para as impressões.
     - Na mensagem, deverá constar o nome e a matrícula da dupla implementadora.
   - Esquema de validação:
     - Coletar os tempos (usando a chamada de sistema) antes e depois da execução de 1 milhão de multiplicações e antes e depois de invocar sleep(5).
     - Para obter um pid, é possível usar a linha de comando (pidof, $$, …) ou usar os valores de pid informado por **ps**.
   - Recomenda-se o uso de distribuição Debian para a execução da tarefa em ambiente virtualizado.

Instruções e Recomendações

A submissão das respostas aos problemas dos trabalhos deverá ser feita através do Moodle da disciplina.

Cada resposta de problema dessa Atividade Extra deverá ser entregue em um pacote ZIP. A dupla de alunos deverá nomear o pacote ZIP da seguinte forma: nome_sobrenome_matricula_nome_sobrenome_matricula_**ae02.**zip.

Entre os artefatos esperados, listam-se:

- códigos-fonte C das soluções dos problemas;
- documentação mínima da aplicação:
  - qual sistema operacional foi usado na construção do sistema;
  - qual ambiente de desenvolvimento foi usado;
  - quais são as telas (instruções de uso)
  - quais são as limitações conhecidas

Não devem ser submetidos executáveis.

Códigos-fonte C com erros de compilação serão desconsiderados (anulados).

Os trabalhos poderão ser realizados em duplas; a identificação de cópia ou plágio irá

provocar anulação de todos os artefatos em recorrência.