Fundamentos de Sistemas Operacionais
Trabalho 01

Prof. Tiago Alves

Introdução aos Sistemas Operacionais

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 ao programador ter profundo conhecimento das interações entre os diferentes *hardwares* que compõem um computador.

Para construir ou adicionar funcionalidades a esses sistemas computacionais, porém, 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) Interagir com ferramentas de desenvolvimento para criação, gerenciamento, depuração e testes de projeto de aplicações.

Referências Teóricas

Capítulos 1 e 2 de Mitchell, Mark, Jeffrey Oldham, and Alex Samuel. Advanced linux programming. New Riders, 2001.

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 usando linguagem C.

Colete o material acompanhante do roteiro do trabalho a partir do Moodle da disciplina e estude os princípios e técnicas de desenvolvimento de aplicações usando linguagem C e sistema operacional Linux.

2) Realizar as implementações solicitadas no questionário do trabalho.

Nas referências bibliográficas que acompanham o trabalho há uma breve apresentação de ferramentas de depuração para o ambiente Linux.

Implementações e Questões para Estudo

1) Escreva um programa em C que, em tempo de execução, receba do usuário 3 coordenadas cartesianas dos vértices de um triângulo e, como saída, calcule e imprima na tela o tamanho dos lados, o perímetro e a área do triângulo. Sua implementação deverá atender os seguintes requisitos:
    - Na sua implementação deverá constar, no mínimo, três arquivos **.h** e três arquivos **.c**.
        - Utilize um arquivo **.h** para definir os seguintes tipos compostos: ponto no plano cartesiano e triângulo, com três pontos no plano cartesiano.
        - Utilize um arquivo **.c/.h** para realizar as operações geométricas: cálculo do comprimento do lado do triângulo, cálculo da condição de existência do triângulo, cálculo do perímetro do triângulo e cálculo da área do triângulo.
        - Utilize um arquivo **.c/.h** com as funções de entrada.
        - Utilize um arquivo **.c** com a função **main()**.
    - Sua implementação deverá ser construída a partir de um **Makefile**.
2) Escreva um programa que ordene uma lista de inteiros. A ordem de ordenação, bem como os inteiros a serem ordenados, deverão ser informados como parâmetros da linha de comando. Na ausência de uma *switch* (opção), a ordem deverá ser crescente; caso o usuário indique a opção **-d**, a ordem deverá ser **crescente** e, se indicar **-r**, a ordem deverá ser decrescente.
    - Na sua implementação deverá constar, no mínimo, dois arquivos **.c** e dois arquivos **.h**.
    - Sua implementação deverá ser construída a partir de um **Makefile**.
    - O vetor usado para armazenar temporariamente os inteiros recebidos pela linha de comando deverá ser alocado dinamicamente e com quantidade de células adequada para a execução do programa.
3) Escreva um programa em C que possua, como variáveis, os seguintes itens:

```
double number1 = 7.3, number2;
char s1[100], s2[100];
```

    - Declare a variável `dPtr` como ponteiro para a variável do tipo double.
    - Carregue o endereço da variável `number1` no ponteiro `dPtr` .
    - Imprima em tela o valor da variável apontada por `dPtr` partindo do ponteiro.
    - Carregue o valor da variável apontada por `dPrt` na variável `number2` .
    - Imprima o valor armazenado em `number2`.
    - Imprima o endereço de `number1` na tela.
    - Imprima o endereço armazenado em `dPtr` na tela .

- O valor impresso decorrente do enunciado que contempla o item anterior é igual ao valor do endereço gravado em `dPrt`?
- Leia uma string e a armazene no array `s1`. Copie a string armazenada em `s1` para `s2`. Compare a string armazenada em `s1` com a string armazenada em `s2` e imprima o resultado na tela.
- Apense a string `s2` à string `s1`. Imprima o resultado na tela.
- A execução do item anterior pode provocar algum erro em tempo de execução?
- Determine o comprimento da string armazenada em `s1` e imprima o resultado na tela.

Instruções e Recomendações

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

As soluções do Trabalho 01 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_**trab01.**zip. Dentro desse pacote, deverá existir os diretórios q01, q02 e q03, que conterão, respectivamente, as soluções às Questões 1, 2 e 3.

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;
  - casos de testes que demonstram que a aplicação contempla o comando do trabalho.

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.