

Thread Lab

Sistemas Hardware-Software

Entrega: 12/06

Em sistemas com interface gráfica é comum precisarmos realizar uma tarefa computacionalmente pesada e ao mesmo tempo manter a interface gráfica responsiva. Uma solução comum neste caso é dividir o trabalho em threads diferentes: uma atualiza a interface enquanto a outra faz o trabalho pesado.

Neste lab iremos implementar um programa gráfico para copiar arquivos, mostrando uma barra de progresso conforme o tamanho do arquivo. Os objetivos deste lab são

1. **Implementar** um programa multi-thread em um contexto realista;
2. **Revisar** as chamadas de Entrada/Saída com arquivos comuns e **pesquisar** chamadas usadas para listar e criar diretórios;
3. **Revisar** o uso de bibliotecas em programas em C.

Parte 1 - Avaliação

A avaliação do lab é baseada na complexidade do programa implementado. Quanto mais funcionalidades forem implementadas maior será sua nota. Também é necessário seguir a risca os requisitos definidos na próxima seção.

Requisitos do funcionamento do programa

Você deverá seguir as seguintes instruções para a criação de seu programa.

- Você deverá entregar um `Makefile` ou um arquivo shell executável que compila seu programa.
- O programa compilado deverá se chamar `uicp`
- Seu programa deverá funcionar de maneira similar ao programa `cp`:
 - `xcp a b` - copia o arquivo `a` para o arquivo `b`, criando-o se ele não existir.
 - `xcp pasta1 pasta2` - copia todos os arquivos da `pasta1` para dentro da `pasta2`.

Conceito D

- O programa mostra uma interface gráfica com uma barra de progresso para a cópia de um arquivo simples;
- O programa **NÃO** suporta a cópia de pastas;
- Alguma das instruções da seção acima não foi seguida;

Conceito C

- O programa permite a de todos os arquivos de uma pasta para outra pasta.
 - Não é necessário fazer a cópia recursiva (pastas dentro da pasta fonte)
 - Seu programa deve checar se a pasta destino existe. Se não existir deve criá-la.
 - Se o caminho destino for de um arquivo e não uma pasta seu programa deverá retornar um erro.
- Sua janela tem um botão *Cancelar*. Este botão para a cópia do arquivo e deleta o arquivo destino.

Conceito B

- O arquivo destino recebe as mesmas permissões do arquivo fonte.
- A interface gráfica contém duas barras de progresso na cópia de pastas
 - A primeira barra mostra a quantidade de arquivos já copiados
 - A segunda barra mostra o status da cópia do arquivo atual
- O trabalho deverá ser entregue via link para um repositório **privado**.

Conceitos B+, A, A+

A implementação de cada uma das funcionalidades abaixo vale 1.0.

- Cópia recursiva de diretórios: se houver uma pasta dentro da pasta fonte ela (e todos os arquivos dentro dela) serão copiados também.
- Caso algum arquivo destino já exista o programa apresenta, graficamente, um prompt perguntando se ele deve ser substituído.
- Além das duas barras de progresso é mostrado um gráfico indicador de quantos bytes por segundo estão sendo copiados.

Parte 2 - Interfaces gráficas em C

Para a parte gráfica do nosso lab usaremos o biblioteca **raylib**. Ela é feita em C puro e muito simples de usar. O código abaixo cria uma janela e desenha um retângulo vermelho no centro.

```
#include "raylib.h"
```

```
int main()
{
    // Initialization
    //-----
    int screenWidth = 800;
    int screenHeight = 450;

    InitWindow(screenWidth, screenHeight, "raylib [core] example - input mouse wheel");

    int boxPositionY = screenHeight/2;

    SetTargetFPS(60);

    // Main game loop
    while (!WindowShouldClose())    // Detect window close button or ESC key
    {
        BeginDrawing();

        ClearBackground(RAYWHITE);
        DrawRectangle(screenWidth/2 - 40, boxPositionY, 80, 80, MAROON);
        DrawText("Texto teste!", 10, 10, 20, GRAY);

        EndDrawing();
        //-----
    }

    // De-Initialization
    //-----
    CloseWindow();    // Close window and OpenGL context
    //-----

    return 0;
}
```

Você deverá baixar a biblioteca *raylib*, compilá-la estaticamente e desenvolver seu programa. Instruções para linux estão disponíveis [no site da biblioteca](#).

Estratégias de pesquisa

Como podemos ver nos objetivos, este lab envolve uma quantidade razoável de pesquisa. Parte de seu objetivo macro no curso é capacitá-los para buscar soluções de maneira independente e de avaliar se essas soluções são adequadas. Para ajudá-los neste fim escrevi abaixo uma lista de passos que vocês podem adotar na pesquisa deste lab.

1. Se você não sabe o nome da função que quer chamar, faça buscas por *POSIX* + funcionalidade desejada.
 - Ex: para listar arquivos de um diretório – POSIX list dir contents
2. Sabendo o nome da função, consulte seu manual
 - Veja também a seção *see also* para funções possivelmente relacionadas e que podem ser úteis.
3. Após ler o manual e ter uma ideia do funcionamento, busque por exemplos.
 - Ex: POSIX opendir examples
4. Faça um exemplo auto contido do que você quer testar.
 - Comece seu exemplo de um arquivo vazio.
 - **Nunca copie código da internet sem entendê-lo.**



Só serão tiradas dúvidas de código que você saiba explicar o que o código deveria fazer. Antes de pedir ajuda leia atentamente a documentação oficial.