Insper

POSIXlab: arquivos e threads

Sistemas de Hardware e Software, 2019/2

Igor Montagner

Entrega: 29/11

Buscador recursivo

Neste projeto iremos implementar um buscador de textos em arquivos. A estratégia de busca usada será

Requisitos de entrega

Neste projeto iremos trabalhar três objetivos de aprendizado:

1. utilização de bibliotecas compartilhadas em programas escritos em C
2. usar funções POSIX para entrada/saída de arquivos
3. trabalhar com threads para isolar o processamento pesado do código de atualização da interface.

Seu trabalho deverá estar dividido em vários arquivos e usar `Makefile` compilar os arquivos e `pkg-config` para encontrar as bibliotecas compartilhadas.

Parte 1 - busca de arquivos

Seu programa deverá realizar a busca por uma string em um diretório. Esta parte de seu trabalho deverá seguir uma interface de programação fixa. Seu programa deverá ter um arquivo `buscador.c` que define

- uma função `GArray *busca_em_arquivo(char *texto, char *nome_arquivo)` que busca por `texto` no arquivo de nome `nome_arquivo` e devolve uma lista de posições em que este texto aparece.
- uma função `GArray *lista_dir(char *dir)` que lista todos os arquivos no diretório `dir` e devolve um `GArray` com os nomes dos arquivos.
- uma função `GHashTable *busca_na_pasta(char *texto, char *dir)` que busca pela string em todos os arquivos da pasta e retorna uma tabela de hash cuja chave é o nome dos arquivos e os valores são listas com as posições que o texto ocorre. Você deverá combinar as duas funções acima.

A abertura do arquivo, as leituras e as buscas deverão ser implementadas usando somente funções POSIX. Ou seja, não vale usar `scanf` e `fgets` ou funções equivalentes da Glib. O mesmo vale para as funções POSIX de listagem de diretórios. Por outro lado, você pode supor que o arquivo cabe na memória.

Seu arquivo `buscador.c` não deverá possuir uma função `main`. O ponto de entrada do seu programa deverá estar em um arquivo `busca.c`. Para compartilhar as definições das funções em `buscador.c` você deverá criar um arquivo `.h` correspondente.

Restrições

A Glib possui funções para trabalhar com arquivos e diretórios. Você não poderá usá-las. Seu trabalho está restrito a funções do padrão POSIX (usadas em aula ou não) e o uso de funções externas implicará em nota máxima **D**. Se ficar em dúvida pergunte ao professor. Você pode usar todas as funções de string da Glib para facilitar sua vida.

Parte 2 - threads

Na segunda parte iremos implementar as buscas nas pastas usando threads. Nosso objetivo será gerar uma thread para cada arquivo na pasta a ser buscada e sincronizar o acesso a estrutura `GHashTable` usada na função `busca_na_pasta`.

Avaliação

O projeto está organizado em conceitos de dificuldade crescente.

Conceito *D*

O programa funciona na linha de comando como abaixo:

```
$> busca arquivo.txt string
Encontrado nas posições: 1, 2, 44, 66, 222

$> busca pasta string
Arquivo 1.txt:
    Encontrado nas posições: 1, 2, 44, 66, 222
Arquivo aa.txt
    Não encontrado
Arquivo bb.txt
    Encontrado nas posições: 22, 33
```

Você poderá passar um arquivo ou uma pasta e o programa deverá agir de acordo.

Conceito *C*

Seu programa deverá lançar uma thread para cada arquivo da pasta e fazer as buscas em paralelo. Ao finalizar, cada thread deverá se incluir na tabela de hash retornada por `busca_na_pasta`. Não se esqueça de checar:

1. se os acessos a tabela de hash estão propriamente protegidos contra concorrência indesejada
2. se o programa sempre acaba ou se é possível uma condição de deadlock

Com o programa básico feito, você deverá fazer as seguintes tarefas (nesta ordem):

Busca recursiva (+1,0)

A função `lista_dir` faz busca recursiva. Ou seja, se ela encontrar uma pasta deverá realizar a busca também nos arquivos desta pasta (e assim recursivamente). A função deverá retornar os caminhos corretamente (concatenados apasta de origem).

Memória e checagem de erros (+1,0)

Seu programa libera toda a memória que aloca e termina sem erros no `valgrind`.

Threading avançado (+2,0)

Seu programa lança no máximo `N` threads e distribui os arquivos a serem examinados para as threads. Se sua distribuição for estática (cada thread recebe um conjunto de arquivos fixo) sua nota será **1,0**. Se for dinâmica (ao acabar de fazer a busca em um arquivo a thread pede um novo arquivo para ser analisado) sua nota será **2,0**.

Uso avançado da Glib (+1,0)

Se a flag de compilação `USAR_GLIB` estiver ativada seu programa deverá usar funções da *GLib* no lugar das chamadas em baixo nível POSIX. Isto inclui listar arquivos em uma pasta, abrir arquivos e fazer buscas nos arquivos.

Interface Gráfica (+1,0)

Foi criada uma interface gráfica em GTK usando GtkBuilder (e Glade). Essa interface é chamada quando o programa é recebido sem argumentos e deve ter duas telas. Na primeira deve ser possível entrar com a string a ser buscada e escolher a pasta a ser analisada. Na segunda será possível ver em quais linhas a string foi encontrada (o arquivo deverá ser mostrado e as ocorrências coloridas). Uma lista das ocorrências serve de índice.