Universidade de Brasília

Faculdade UnB Gama (FGA)

Engenharia de Software

Fundamentos de Sistemas Operacionais

Professor: Tiago Alves

Trabalho 4

Eduardo Q. Gomes 14/0137068

Miguel Pimentel 14/0156143

Brasília, 22 de Outubro de 2016

Trabalho 4 Fundamentos de Sistemas Operacionais

Material Usado/Ambiente de Desenvolvimento

Para o desenvolvimento deste trabalho foi utilizado como sistema Operacional:

- Sistema Operacional Linux, distribuição Ubuntu 16.04 com a interface Gnome;

Além disso, foi utilizado como ambiente de desenvolvimento:

- Compilador gcc (5.4.0);

- Editor de texto Sublime;

Conteúdo Utilizado

- Conteúdo do Moodle disponível em http://aprender.ead.unb.br/course/view.php?id=3806
- Livro Autor: Autor: SILBERSCHATZ, A.; GAGNE, G.; GALVIN, P.B. Obra: Operating System Concepts
- CodeSourcery, Mark L. Mitchell, Alex Samuel, Jeffrey Oldham-Advanced Linux Programming-New Riders. 2001.

Questões

1) Escreva um programa em C que seja capaz de alterar os metadados de um arquivo.
Requisitos:

O programa deverá receber como entradas o nome de um arquivo e uma string numérica no formato AAAAMMDDHHmm em que A representa dígitos do ano, M, dígitos do mês, D, dígitos do dia, H, dígitos hora e m, dígitos do minuto.

Como saída, o programa deverá:

Gerar uma cópia de backup do arquivo (extensão .bkp) para efeito de validação das alterações realizadas nos metadados de um arquivo.

Imprimir na tela quais são os metadados referentes ao horário da criação, modificação e acesso do arquivo alvo.

Alterar os metadados do arquivo alvo de forma que as informações de horário sejam substituídas pelo valor informado na string AAAAMMDDHHmm.

Instruções de Uso

A questão 1 se encontra disponível no diretório q01. Ressalta-se, o fato das instruções de uso do programa estarem presentes no **README.md**.

De forma geral, para executar:

- Abra o Terminal para a maioria das distros linux: ctrl + alt + t e entre no diretório q01.
- Para **compilar**, digite:

  make

- Para **executar**, digite:

  ./q01 <file_name> <YYYYMMDDHHmm”

O parâmetro de entrada <file_name> representa o arquivo que seja desejado gerar o backup(“bkp”) e modificar os metadados do arquivo. O segundo parâmetro representa os dados de data e horário que serão alterados no arquivo escolhido. A string <YYYYMMDDHHmm> se refere a um uma data e horário da seguinte forma: YYYY representa o ano; MM representa o mês; DD representa o dia ; HH representa a hora; e mm representa os minutos.

- Para **excluir** o executável:

  make clean

Limitações Conhecidas

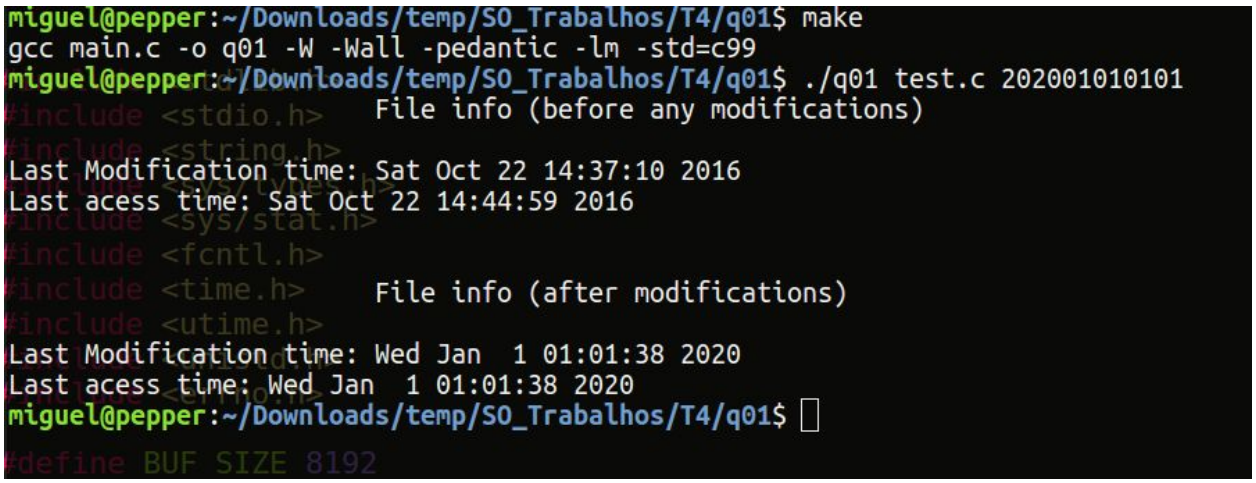
Não foi possível acessar e consequentemente acessar o birth time do arquivo. Entretanto, quando o “bkp” é criado esta informação fica disponível como última modificação.

Caso de Testes e Telas de Uso

No diretório q01 está disponível dois casos de teste:

1. Caso de Teste 1:

./q01 test.c 202001010101



2. Caso de Teste 2:

./q01 test2.c 201101091820

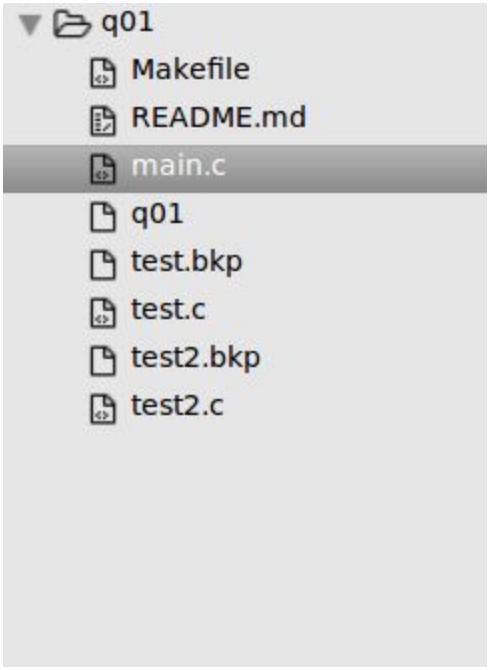
```
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T4/q01$ ./q01 test2.c 201101091820
                    File info (before any modifications)

Last Modification time: Sat Oct 22 14:37:59 2016
Last acess time: Sat Oct 22 14:44:59 2016

                    File info (after modifications)

Last Modification time: Sun Jan  9 18:20:33 2011
Last acess time: Sun Jan  9 18:20:33 2011
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T4/q01$
```

Backups criados com a execução de ambos



2) Escreva um programa em C chamado “buscador” que contemple os seguintes requisitos:

O programa receberá 3 argumentos: o primeiro argumento será o caminho dentro da árvore de diretórios do sistema.; segundo argumento será uma string de caracteres; o terceiro argumento será N, o número máximo de linhas de saída a serem geradas.

Durante sua execução, buscador deverá registrar, em saída, todos os arquivos que contenham, em seu nome, uma substring igual ao segundo argumento.

O seu programa deverá mostrar, no máximo, N entradas na árvore de diretórios, e, cada linha de saída, deverá ser composta da seguinte forma: o primeiro argumento é a raiz da estrutura de diretórios de onde se iniciou a busca; e o segundo argumento é uma substring com caminho do arquivo cujo nome contenha o segundo argumento, a terceira parte, será composta pelos 30 primeiros bytes do arquivo.

Instruções de Uso

Para executar siga as instruções:
Ressalta-se, o fato das instruções de uso do programa estarem presentes no README.md.

De forma geral, para executar:

- Abra o Terminal para a maioria das distros linux: ctrl + alt + t.
- Entre dentre uma das versões implementadas;
- Para **compilar**, digite:

  make

- Para **executar**, digite

  make run

Se preferir, já está pré definido um teste em run que pode ser alterado da maneira que desejar

Por exemplo:

```
./buscador ./archievesOfTest test 7
./buscador ./archievesOfTest test 10
./buscador . test 7
./buscador . 12 7
```

- Para **excluir** o executável:

make clean

Limitações Conhecidas

As entradas devem estar restritamente escritas no terminal dessa forma:

./buscador <path> <string> <max_lines>

Onde ./buscador é o executável, path é o local onde irá executar, string é a substring que quer encontrar e max_lines é a quantidade de pesquisas que deseja encontrar.

Não foram reconhecidas limitações quanto a execução do programa

Resposta Questão do Trabalho

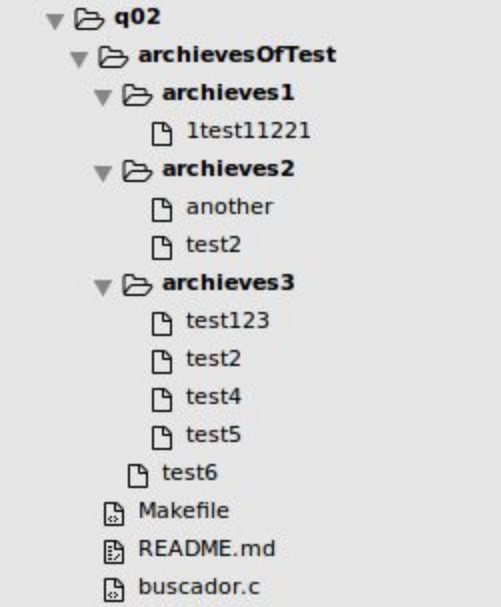
Descreva o que é Filesystem Hierarchy Standard (FHS) e indique qual é a destinação típica das pastas tipicamente encontradas na raiz de um sistema de arquivos UNIX.

Filesystem Hierarchy Standard (FHS ou padrão para sistema de arquivos hierárquico) tem como propósito definir os principais diretórios e conteúdos de um SO do tipo UNIX.
As principais pastas são pastas tipicamente encontradas são:

- /bin que tem os comandos binários principais para todos os usuários;
- /boot que tem por exemplo o boot loader, arquivos de boot;
- /dev é relacionado a dispositivos;
- /etc arquivos de configuração do computador;
- /home contém os diretórios dos usuários, opcional;
- /lib bibliotecas essenciais para arquivos de /bin e /sbin;
- /opt tem os pacotes estáticos para aplicações;
- /root é o diretório home para o super usuário;
- /tmp contém arquivos temporários;
- /usr arquivos compartilhados para os usuários, apenas leitura.

Caso de Testes e Telas de Uso

Para executar os testes foi utilizada uma árvore de diretórios teste:



Casos de teste na próxima página.

Caso de Teste: ./buscador ./archievesOfTest test 7

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4$ cd q02/
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$ make
gcc buscador.c -o buscador -pedantic -lm
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$ ./buscador ./archievesOfTe
st test 7
Origem: ./archievesOfTest
Nome da substring: test
Quantidade de arquivos a serem lidos: 7

Arquivo encontrado: ./archievesOfTest/archieves3/test123
A prática cotidiana prova que

Arquivo encontrado: ./archievesOfTest/archieves3/test2
No mundo atual, a mobilidade d

Arquivo encontrado: ./archievesOfTest/archieves3/test4
Acima de tudo, é fundamental

Arquivo encontrado: ./archievesOfTest/archieves3/test5
Todavia, o desenvolvimento con

Arquivo encontrado: ./archievesOfTest/archieves2/test2
No mundo atual, a mobilidade d

Arquivo encontrado: ./archievesOfTest/test6
No mundo atual, a percepção

Arquivo encontrado: ./archievesOfTest/archieves1/1test11221
Pensando mais a longo prazo, o

eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$
```

Caso de Teste: make run
Como já dito, para auxiliar na rapidez do manuseio da execução há um make run no makefile.

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$ make run
./buscador ./archievesOfTest test 7
Origem: ./archievesOfTest
Nome da substring: test
Quantidade de arquivos a serem lidos: 7

Arquivo encontrado: ./archievesOfTest/archieves3/test123
A prática cotidiana prova que

Arquivo encontrado: ./archievesOfTest/archieves3/test2
No mundo atual, a mobilidade d

Arquivo encontrado: ./archievesOfTest/archieves3/test4
Acima de tudo, é fundamental

Arquivo encontrado: ./archievesOfTest/archieves3/test5
Todavia, o desenvolvimento con

Arquivo encontrado: ./archievesOfTest/archieves2/test2
No mundo atual, a mobilidade d

Arquivo encontrado: ./archievesOfTest/test6
No mundo atual, a percepção

Arquivo encontrado: ./archievesOfTest/archieves1/1test11221
Pensando mais a longo prazo, o

eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$
```

Caso de Teste: ./buscador . 12 4

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$ make
gcc buscador.c -o buscador -pedantic -lm
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$ ./buscador . 12 4
Origem: .
Nome da substring: 12
Quantidade de arquivos a serem lidos: 4

Arquivo encontrado: ./archievesOfTest/archieves3/test123
A prática cotidiana prova que

Arquivo encontrado: ./archievesOfTest/archieves1/1test11221
Pensando mais a longo prazo, o
```

Caso de Teste: ./buscador . test 3

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$ make
gcc buscador.c -o buscador -pedantic -lm
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T4/q02$ ./buscador . test 3
Origem: .
Nome da substring: test
Quantidade de arquivos a serem lidos: 3

Arquivo encontrado: ./archievesOfTest/archieves3/test123
A prática cotidiana prova que

Arquivo encontrado: ./archievesOfTest/archieves3/test2
No mundo atual, a mobilidade d

Arquivo encontrado: ./archievesOfTest/archieves3/test4
Acima de tudo, é fundamental
```