



Universidade de Brasília

Faculdade UnB Gama (FGA)

Engenharia de Software

Fundamentos de Sistemas Operacionais

Professor: Tiago Alves

Trabalho 3

Eduardo Q. Gomes 14/0137068

Miguel Pimentel 14/0156143

Brasília, 28 de Setembro de 2016

Trabalho 1 Fundamentos de Sistemas Operacionais

Material Usado

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;
- Valgrind;
- Gdb;

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.
- <<http://br.ccm.net/faq/10300-verificar-se-um-inteiro-e-um-numero-primo-em-c>> acesso em 28 de Setembro de 2016.
- <<http://mindbending.org/pt/bibliotecas-estaticas-simples>>, acesso em 28 de Setembro de 2016.
- <<http://www.ibm.com/developerworks/br/library/l-dynamic-libraries/>>, acesso em 29 de Setembro de 2016.

Questões

As questões encontram-se organizadas da seguinte maneira:

- diretório - versao_estatica: Programa que gera números primos e testa números primos, compilado usando a versão estática da biblioteca (questão 1a);
- diretório - versao_dinamica: Programa que gera números primos e testa números primos, compilado usando a versão dinâmica da biblioteca(questão 1b);
- diretório - versao_carga_sobre_demanda: programa que gera números primos e testa números primos, porém com carga sobre demanda, usando os serviços providos pela libdl (dlfcn.h) (questão 1c);

Respostas Questões do Trabalho

Em relação ao programa que contempla os itens a) e b), quais foram as alterações de códigos-fonte necessárias para a solução (se houverem)?

R:

Para os códigos em si, os arquivos main.c primo.h é primo.c não foram alterados, porém a arquitetura de pastas foi alterada para exercitar mais os conceitos de makefile e como referenciar a partir dele, o que possibilitou entender melhor seu funcionamento.

Em relação ao código, para fazer as bibliotecas e executar foi necessário alterações nos makefiles, pois as bibliotecas foram feitas de forma diferente.

Dados os conhecimentos adquiridos em função desse trabalho, indique vantagens e problemas decorrentes da utilização de bibliotecas dinâmicas.

R:

A partir do conhecimento adquirido algumas conclusões foram possíveis serem feitas acerca de bibliotecas.

Em bibliotecas dinâmicas o uso de é dado em tempo de execução, vários programas podem usar as funções da mesma biblioteca, o que otimiza a utilização do espaço de memória. Porém as bibliotecas dinâmicas somente são carregadas quando um aplicativo é carregado e a ligação ocorre no momento da execução. Um risco muito grande envolvido com esse tipo de biblioteca é se uma biblioteca dinâmica tem seu nome alterado, é removida ou movida, todos os programas relacionados a ela podem ter um mal funcionamento.

Bibliotecas estáticas têm vantagens em contextos simples, pois exige funcionalidade mínima, porém cada programa tem sua própria biblioteca, causando um certo tipo de redundância.

Questão 1a - versao_estatica

Instruções de Uso

Para a utilização de cada uma das implementações entre no diretório que representa esta opção conforme descrito anteriormente em questões. Por exemplo, para utilizar a versão estática entre no diretório : `versao_estatica`. 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 dentro uma das versões implementadas;
- Para **compilar**, digite:

`make`
- Para **executar**, digite:

`make run`
- Para **excluir** o executável:

`make clean`

Destaca-se novamente a presença do **README.md** caso tenha alguma dúvida.

Limitações Conhecidas

A função `rand()` realiza um sorteio pseudo-aleatório. Entretanto, se o executável for executado várias vezes num intervalo de tempo, pode ser que a saída seja a mesma, contrariando a probabilidade.

Caso de Testes e Telas de Uso

Foi utilizado como verificador o site <http://www.onlineconversion.com/prime.htm>, para validar as entradas de cada caso de teste.

```
miguel@pepper:~/Downloads/temp/S0_Trabalhos/T3/versao_estatica$ make
#referencia: http://mindbending.org/pt/bibliotecas-estaticas-simples
rm build/*
gcc -c lib/primo.c -o build/primo.o
ar rcs build/libprimo.a build/primo.o
ar -t build/libprimo.a
primo.o
nm -s build/libprimo.a

Archive index:
gera_primo in primo.o
testa_primo in primo.o

primo.o:
0000000000000000 T gera_primo
                 U rand
                 U sqrt
                 U srand
0000000000000034 T testa_primo
                 U time
cp lib/primo.h build/primo.h
gcc -static src/main.c -L./build -I./build -lprimo -o build/main.run -W -Wall -pedantic -ansi -lm
./build/main.run
Numero primo: 1652029661
miguel@pepper:~/Downloads/temp/S0_Trabalhos/T3/versao_estatica$ make run
./build/main.run
Numero primo: 1162562483
miguel@pepper:~/Downloads/temp/S0_Trabalhos/T3/versao_estatica$ make clean
rm build/*.o build/main.run
miguel@pepper:~/Downloads/temp/S0_Trabalhos/T3/versao_estatica$
```

Os valores descritos em “Número primo” na imagem acima são primos. Assim, o programa atende ao comportamento esperado, ou seja, apresentar números primos de forma aleatória.

Questão 1b - versao_dinamica

Instruções de Uso

Para a utilização de cada uma das implementações entre no diretório que representa esta opção conforme descrito anteriormente em questões. Por exemplo, para utilizar a versão estática entre no diretório : versao_estatica.

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 dentro uma das versões implementadas;
- Para **compilar**, digite:

make

- Para **executar**, digite

make run

- Para **excluir** o executável:

make clean

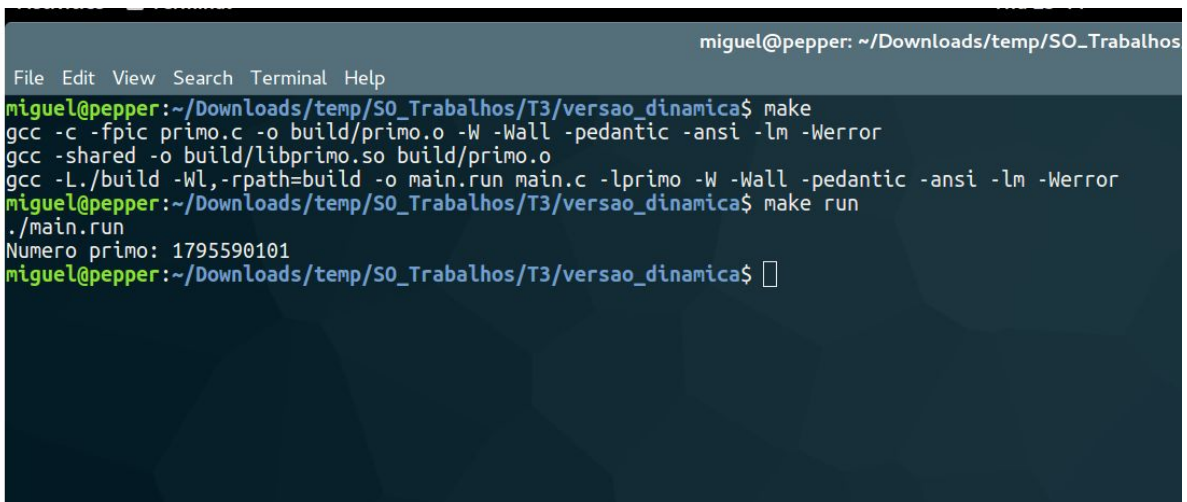
Destaca-se novamente a presença do README caso tenha alguma dúvida.

Limitações Conhecidas

A função rand() realiza um sorteio pseudo-aleatório. Entretanto, se o executável for executado várias vezes num intervalo de tempo, pode ser que a saída seja a mesma, contrariando a probabilidade.

Caso de Testes e Telas de Uso

Foi utilizado como verificador o site <http://www.onlineconversion.com/prime.htm>, para validar as entradas de cada caso de teste.



```
miguel@pepper: ~/Downloads/temp/SO_Trabalhos
File Edit View Search Terminal Help
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T3/versao_dinamica$ make
gcc -c -fpic primo.c -o build/primo.o -W -Wall -pedantic -ansi -lm -Werror
gcc -shared -o build/libprimo.so build/primo.o
gcc -L./build -Wl,-rpath=build -o main.run main.c -lprimo -W -Wall -pedantic -ansi -lm -Werror
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T3/versao_dinamica$ make run
./main.run
Numero primo: 1795590101
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T3/versao_dinamica$
```

Os valores descritos em “Número primo” na imagem acima são primos. Assim, o programa atende ao comportamento esperado, ou seja, apresentar números primos de forma aleatória.

Questão 1c - versao_carga_sobre_demanda

Instruções de Uso

Para a utilização de cada uma das implementações entre no diretório que representa esta opção conforme descrito anteriormente em questões. Por exemplo, para utilizar a versão estática entre no diretório : versao_estatica.

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 dentro uma das versões implementadas;

- Para **compilar**, digite:

```
make
```

- Para **executar**, digite:

```
make run
```

- Para **excluir** o executável, digite:

```
make clean
```

Destaca-se novamente a presença do READ.md caso tenha alguma dúvida.

Limitações Conhecidas

A função rand() realiza um sorteio pseudo-aleatório. Entretanto, se o executável for executado várias vezes num intervalo de tempo, pode ser que a saída seja a mesma, contrariando a probabilidade. Além disso, o rand gera número maiores que 32767 no geral.

Caso de Testes e Telas de Uso

Foi utilizado como verificador o site <http://www.onlineconversion.com/prime.htm> , para validar as entradas de cada caso de teste.



```
Activities Terminal Thu 15:45
miguel@pepper: ~/Downloads/temp/SO_Trabalhos/T3
File Edit View Search Terminal Help
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T3/versao_carga_sobre_demanda$ make
gcc -c -fpic primo.c -o build/primo.o -W -Wall -pedantic -ansi -lm
gcc -shared -o build/libprimo.so build/primo.o
gcc -L./build -Wl,-rpath=build -o main.run main.c -lprimo -ldl -W -Wall -pedantic -ansi -lm
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T3/versao_carga_sobre_demanda$ make run
./main.run
Numero primo: 502083397
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T3/versao_carga_sobre_demanda$ make clean
rm main.run
miguel@pepper:~/Downloads/temp/SO_Trabalhos/T3/versao_carga_sobre_demanda$
```

Os valores descritos em “Número primo” na imagem acima são primos. Assim, o programa atende ao comportamento esperado, ou seja, apresentar números primos de forma aleatória.