

# FSO - Trabalho 3, Questão 1, Item B

---

Alunos: Dylan Guedes e Artur Bersan

Matrículas: 12/0115727 e 14/0016813

## Sistema Operacional Utilizado

---

Ubuntu Linux - x86\_64 Kernel 3.13.0-32-generic

Linux Fedora 24 - x86\_64 Kernel 4.6.3-300

## Ambiente de Desenvolvimento

---

Computadores pessoais. Utilizamos o Linux, Makefile, gcc, pthread, git, valgrind e vim.

## Questões

---

Quais foram as alterações de códigos-fonte necessárias para a solução (se houverem)?

Não foram necessárias quaisquer alterações no código fonte da biblioteca ou do código que utiliza a biblioteca. A diferença ficou na maneira com que o binário e os arquivos da libprimo foram gerados.

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

Por conta do escopo do trabalho, não foram percebidas grandes diferenças entre utilizar a abordagem estática ou dinâmica para a biblioteca criada (libprimo). Contudo, tem-se noção de que as bibliotecas dinâmicas apresentam as seguintes vantagens:

- É possível trocar componentes utilizados pela biblioteca `.so` sem que seja necessário recompilar todo o restante do código
- O executável apresenta um tamanho menor, por ter menos código repetido

E as seguintes desvantagens:

- Algumas fases terão performance relativamente pior que a abordagem estática, pois um overhead adicional aparece na fase de loading
- Enquanto na abordagem estática o binário já contém todas as dependências, na abordagem dinâmica é preciso ter todas as dependências instaladas/a disposição. Ou seja, uma desvantagem é o fato de ter várias dependências separadas, e precisar que elas sempre estejam presentes (enquanto na abordagem estática bastava só ter o arquivo binário)

# Instruções

---

## Compilando

Para compilar utilizando a versão dinâmica da biblioteca, basta rodar o comando:

```
$ make all
```

---

Para remover os arquivos temporários e os executáveis

```
$ make clean
```

## Utilização

Após ser compilado, o binário pode ser executado através do comando

```
$ ./app
```

Um número primo aleatório será então gerado e mostrado na tela, e será analisado se um dado número é primo (e é), e se outro dado número não é primo (e não é).

## Caso de Teste Válido

O usuário não interage com a aplicação. Embora os resultados do número primo gerado mudem, pois é aleatório, o usuário não tem poder sobre o resultado que será apresentado.

## Limitações conhecidas

---

Números primos maiores que 50000 não podem ser gerados. A função `rand()` é pseudoaleatória, se o binário for executado várias vezes em um curto intervalo de tempo o resultado vai ser o mesmo.