

**EA872 Laboratório de Programação de Software Básico**  
**Atividade 9**



**Vinícius Esperança Mantovani**

**RA 247395**

Entrega (limite): 20/10/2023, 13:00.

### **Introdução:**

Nesta atividade, adicionou-se ao servidor o atendimento a requisições por diferentes subprocessos, gerados por chamada “fork()” e controlados precisamente por uma variável “cont\_processos”, responsável por armazenar o número de processos criados. Desse modo, controla-se tanto o número de processos criados, como, equivalentemente, o número de requisições sendo atendidas simultaneamente. Neste contexto, pode-se implementar com sucesso um sistema de atendimento de requisições de maneira independente entre uma e outra, com limite estabelecido de requisições simultâneas dado pela variável “max\_procs”.

Ademais, pode-se também, garantir o controle de erros de código “EINTR”, ocasionados por interrupções durante a execução da chamada “accept()”. E, pôde-se garantir o tratamento adequado para os casos de erro de excesso de requisições simultâneas.

### **Explicações precedentes aos testes:**

#### **- Quanto ao modo e aos ajustes:**

Deve-se notar que todos os testes são referentes às respostas a requisições distintas, sejam elas simultâneas ou não, para garantir tanto a corretude das respostas em caso de análises simultâneas ou individuais como, também, a corretude delas em casos de excesso de requisições em um mesmo momento. Para que tal análise fosse possível, fez-se que o número de requisições simultâneas estabelecido como máximo fosse baixo, sendo este de 2 para a execução dos testes e, fez-se também que as requisições demorassem uma quantia razoável de tempo para serem satisfeitas, usando-se chamadas “sleep(5)”, para possibilitar o processamento de mais de uma chamada em um mesmo momento.

#### **- Quanto ao que se apresenta nas imagens e nos textos:**

Nas imagens que se seguem, todas referentes aos testes feitos, tem-se uma ou mais guias amostradas e elas representam respostas às requisições, feitas de maneira simultânea ou não. Aquelas que foram feitas de maneira individual são expressas em imagens numeradas mas não “subnumeradas” em “a”, “b”... Já aquelas que apresentam testes de simultaneidade,

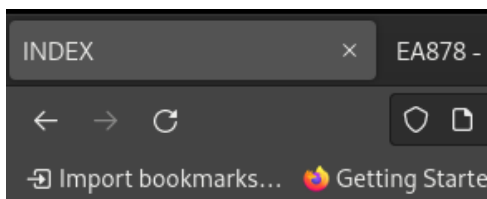
são também numeradas, mas apresentam conjuntamente “subnumerações” em letras, para mostrar que são diferentes guias que expressam respostas a requisições feitas simultaneamente.

Já os textos apresentados, se referem às respostas induzidas no terminal por meio de chamadas de função “printf()” responsáveis por imprimir os valores de “cont\_processos” em diferentes posições e tempos no código. Dessa maneira, para entender-se corretamente o que se apresenta nos textos dos testes, deve-se ter sob conhecimento que as linha com:

- 1) “numero de processos no else: <cont\_processos>”: apresentam o número de subprocessos em aberto durante o início do atendimento de uma nova requisição (incluindo o processo associado a esta nova requisição).
- 2) “numero de processos no if: <cont\_processos>”: apresentam o número de subprocessos em aberto durante uma entrada no bloco de tratamento de excesso de requisições simultâneas e, apresentam neste código, portanto, o valor sempre igual a 2 (máximo de requisições permitidas simultaneamente usado para os testes) para mostrar que, de fato, o número seria excedido caso um novo processo-filho fosse criado.
- 3) “numero de processos no tratamento: <cont\_processos>”: apresentam o número de subprocessos em aberto quando a função de tratamento de sinais de encerramento de processos-filho (“morre\_processo()”) é chamada, considerando já a redução em um no número de subprocessos em execução no momento.

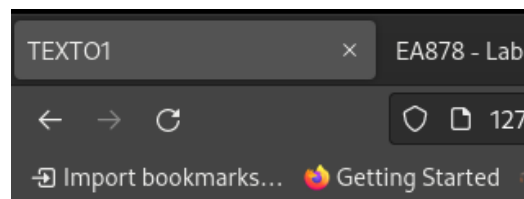
### Testes do programa:

A seguir, são amostradas respostas a requisições feitas individualmente sem concorrência, ou seja, sem simultaneidade:



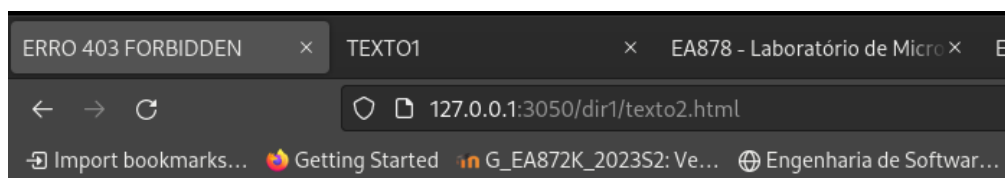
ESTE É O ARQUIVO INDEX

Figura 1: Resposta à requisição por arquivo “index.html”



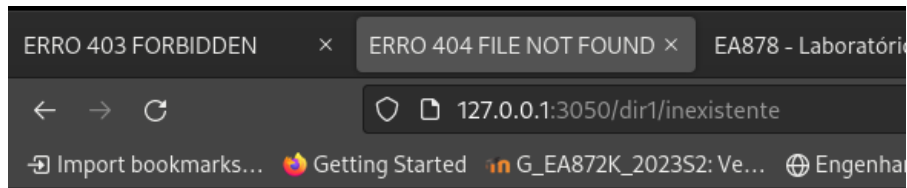
ESTE É O ARQUIVO TEXTO 1

Figura 2: Resposta à requisição por arquivo “texto1.html”



Um erro foi identificado: O arquivo nao possui permissao para leitura!

Figura 3: Resposta à requisição por arquivo “texto2.html”



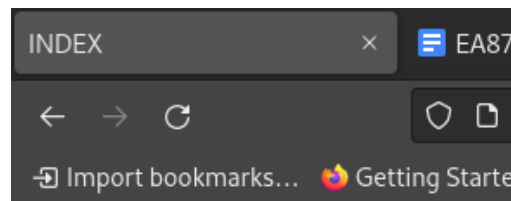
Um erro foi identificado: O arquivo requisitado não existe!

Figura 4: Resposta à requisição por arquivo “inexistente”

```
Unset
numero de processos no else: 1
numero de processos no tratamento: 0
```

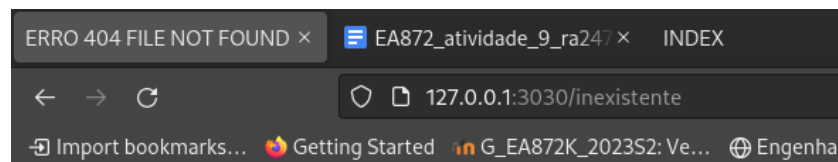
Texto 1: retorno no terminal para todos os casos apresentados acima (Figuras 1 a 4)

Agora, apresenta-se a seguir testes para casos de requisições simultâneas.



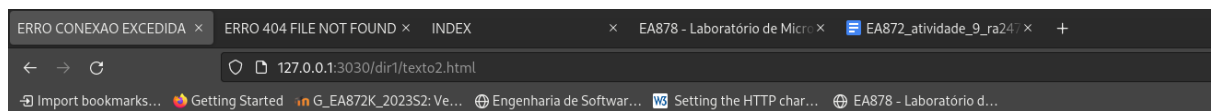
ESTE É O ARQUIVO INDEX

Figura 5: Resposta à requisição por arquivo “index.html”



Um erro foi identificado: O arquivo requisitado não existe!

Figura 6: Resposta à requisição por arquivo “inexistente”



Um erro foi identificado: O servidor está indisponível, o número máximo de requisições simultâneas foi alcançado, tente novamente!

Figura 7: Resposta à requisição por arquivo “texto2.html” excedendo número máximo de requisições simultâneas permitido

```
Unset
numero de processos no else: 1
numero de processos no else: 2
numero de processos no if: 2
numero de processos no tratamento: 1
```

```
numero de processos no tratamento: 0
```

Texto 2: retorno para a sequência de requisições chamadas simultaneamente.

No texto acima, pode-se notar que o número de requisições (processos-filhos) foi aumentado em 1 duas vezes (`numero de processos no else: ...`) e, na tentativa de requisição terceira, ocorre uma entrada no bloco “if” que trata do caso de excesso de requisições simultâneas e, logo na entrada deste bloco, existem dois processos-filhos em execução, como se esperava. Por fim, temos duas linhas impressas pela função de tratamento de sinal de encerramento de processos-filhos (`numero de processos no tratamento: ...`), mostrando dois decrementos subsequentes no valor da variável “cont\_processos”, o que demonstra a finalização de dois processos-filhos (aqueles criados para atender às duas requisições iniciais).