48.255-2 – Laboratório de Sistemas Operacionais

Turmas A e B

2015/2

Projeto 4: Programação Concorrente

1 Objetivo

Aprender com um exemplo mais concreto a usar os mecanismos básicos de programação concorrente.

2 Tarefas

A tarefa principal neste projeto é incluir suporte a múltiplas threads em um programa que quebra senhas fornecido pelo professor. Mais precisamente, o programa procura a partir de um *hash* de senha (como encontrado no arquivo `/etc/shadow`) a senha que foi usada para criar este *hash*. O programa exemplo é bem simples e quebra senhas de exatamente 4 caracteres, cifradas usando a função `crypt()` padrão de Unix (3DES), por meio de busca exaustiva. Como cada teste é independente do demais, é fácil usar várias threads (e CPUs).

As atividades a serem cumpridas são:

1. Baixar, compilar e executar o programa exemplo fornecido. Estudar o seu funcionamento.
2. Medir o tempo que este programa leva para quebrar o hash "aaRCVPtrkrWUY".
3. Alterar o programa, adicionando suporte para múltiplas threads.
4. Medir o tempo que o programa *multithreaded* leva para quebrar o hash "aaRCVPtrkrWUY".

3 Entregas

Você deve entregar:

- O código do seu programa *multithread*.
- Um relatório descrevendo a estratégia que você empregou para dividir o problema de quebrar a senha em múltiplas threads, e como você implementou esta estratégia. Não esqueça de informar os tempos obtidos na execução com uma e com múltiplas threads.

Os projetos são em duplas. As entregas devem ser feitas via Moodle, em dois arquivos distintos: o código do seu programa *multithread* e seu relatório em formato PDF.

4 Dicas

Uma forma muito interessante, e fortemente recomendada, de se organizar um programa *multithreaded* como este é a organização produtor/consumidor. Os códigos mostrados em sala de aula exemplificando este padrão podem ser usados como base para construir a sua solução. Estes exemplos usando pthreads estarão disponíveis no Moodle.

Para testar o seu programa, use o seguinte comando para gerar *hashes* de teste:

```
$ perl -e 'print crypt("senha-de-4-caracteres", "aa") . "\n"'
```

Pode ser interessante também modificar o programa para usar senhas de 3 ou 2 caracteres durante os testes.

Para medir o tempo de execução de um programa utilize o comando `time`:

```
$ time ./crack-passwd aaRCVPtrkrWUY
```

Por mais interessante que seja, este nosso quebrador de senhas não tem aplicação prática em seu formato atual por várias razões. Um usuário comum não consegue ler os hashes em `/etc/shadow` para tentar quebrá-las. Nenhuma distribuição Linux usa senhas DES hoje em dia, a maioria usa senhas SHA-1 ou SHA-512, que são bem mais lentas para gerar. Por fim, nem os usuários mais bobos do mundo usam senhas de apenas quatro caracteres.

Para os curiosos, no Moodle podem ser achados referências para dois artigos interessantes sobre o assunto.