A Cifra de César

Universidade Tecnológica Federal do Paraná – Campus Ponta Grossa
Disciplina de Arquitetura e Organização de Computadores – CC33B
Rafael Althaus Capri Castelo Branco Lisboa
RA: 1591851

1. Introdução

O termo criptografia surgiu das palavras gregas "*kryptós*" (oculto) e "*gráphein*" (escrita). Nada mais é do que um conjunto de técnicas e instruções que visam ocultar a informação contida em uma palavra, frase ou qualquer outro agrupamento de carácteres. Utilizada desde os primórdios da civilização moderna, principalmente para fins militares, mensagens eram criptografadas para que o inimigo não fosse capaz de descobrir a informação, caso se apoderasse dela. Uma das primeiras técnicas de cifra encontra-se entre 600 a.C. e 500 a.C., quando os Hebreus utilizaram-se de uma instrução de substituição simples, onde os caracteres são trocados um a um por outros, e com ela escreveram o Livro de Jeremias.

Porém, o método de criptografia mais conhecido é o "Cifra de César". Utilizados por generais Romanos em meio a guerra para enviar e receber mensagens, a cifra utilizava-se de um sistema muito simples: bastava um carácter do alfabeto ou numérico 3 casas a frente, e fazer o processo reverso para descriptografar. O método tornou-se inútil após o inimigo ter descoberto como funcionava o sistema, mas ainda sim rendeu muitas vitórias a César e seus generais.

Hoje, a Cifra de César é extensamente estudada como um exemplo básico e de simples entendimento no ensino de criptografia, principalmente para alunos de cursos relacionados à segurança de informações, como a área da computação.

A cifra a qual nos referimos acima será o objeto de estudo desse trabalho, onde se foi desenvolvido um programa na linguagem Assembly para processadores de arquitetura MIPS, cujo objetivo é de cifrar e decifrar mensagens utilizando-se da "Cifra de César".

2. Desenvolvimento

Desenvolvido no simulador MIPS e IDE MARS 4.5, utilizando-se da linguagem Assembly, o programa consiste em uma série de instruções que guiam o usuário para cifrar ou decifrar uma mensagem.

Iniciamos o software oferecendo ao usuário duas opções principais: a de ler instruções e mensagem diretamente de um arquivo de extensão TXT ou deixar que elas sejam inseridas manualmente.

Vamos começar supondo que a primeira opção foi escolhida. O programa armazena esse número e abre um input para que o usuário digite o nome (com extensão) do arquivo que contém as instruções e a mensagem. O arquivo, do tipo TXT, deverá conter as seguintes informações, na ordem apresentada:

- Um dígito inicial ("1" ou "2") para cifrar ou decifrar, respectivamente, seguido de um espaço.
- Dois dígitos, sendo esses relacionados ao fator da cifra, ou seja, quantas casas serão utilizadas para cifrar ou decifrar a mensagem. O programa não trabalha com dígitos maiores que 99 (explicação para tal circunstância na sessão "Resultados e Troubleshooting"). Caso o dígito seja menor que 10, deve-se acrescer um "0" a frente do mesmo.

- A mensagem a ser decodificada.

Exemplo: *1 05 O rapaz de camisa branca entrou em seu carro.*

Após o nome do arquivo ser recebido, esse é armazenado em uma *String*, cujo carácter final (0xa0) é removido para que a leitura possa ocorrer sem problemas. Logo, o arquivo é aberto através de uma chamada de sistema e seu conteúdo armazenado em uma *String* dentro do programa, para que este seja utilizado no decorrer do processo. O primeiro carácter que regula a opção de cifra/decifra é lido e, logo após, a opção de fator. Ambos são armazenados em

registradores para serem utilizados mais a frente. A *String* do texto é então limpa, retirando os cinco primeiros bytes que diziam respeito às instruções e, como estes já foram armazenados, já não mais nos interessam e podem ser eliminados. Para isso, o programa apenas coloca um "0" no lugar dos cinco primeiros bytes, e avança o endereço de memória do texto em cinco posições, este iniciando agora na mensagem. A partir daqui, inicia-se o processo de cifra.

Voltando ao início do programa, caso a segunda opção tenha sido escolhida, ou seja, o usuário gostaria de digitar as instruções e mensagens manualmente. Após "2" ter sido escolhido, duas novas opções aparecem, a de criptografia e descriptografia. Escolhida qualquer uma das duas, o programa pede ao usuário que digite o texto a ser tratado. Feito isso, o fator é requisitado. Nota-se que até aqui, as duas opções não se diferem muito quanto ao processo de recebimento dos valores a serem utilizados. Em ambos os casos, a opção de cifra/decifra, o fator e o texto são armazenados nos mesmos registradores, passando ambos pelo mesmo processo de criptografia, divergindo novamente apenas no final do programa.

Agora que conhecemos as duas opções de entrada de dados e possuímos todos os valores registrados e prontos para iniciar o processo de criptografia, vamos entender como o código se comporta ao tratar a *String* da mensagem. Iniciamos lendo o valor da opção de cifra/decifra. Caso seja "1", o programa é instruído a avançar o número de casas do fator, caso "2", o número do fator é negativado, causando o código a efetuar o processo reverso da criptografia. Logo após, iniciamos armazenando o primeiro carácter da mensagem, esse lido conforme seu código na tabela ASCII. Esse então é comparado com os intervalos da tabela, para verificar se ele se trata de uma letra maiúscula, minúscula, números ou se encontra em um dos quatro intervalos de caracteres especiais. Quando esbarrado em um dos intervalos, este carácter então é adicionado/subtraído o número do fator, sendo armazenado na sua posição e fazendo com que o programa retorne ao início do loop, utilizando-se do próximo carácter até que todos estejam devidamente tratados.

Após sua saída do loop, o programa verifica se as instruções foram entradas manualmente ou através de arquivo, utilizando-se de uma comparação com o registrador onde essa informação foi armazenada. Logo, dois caminhos são possíveis: o de imprimir na tela, ou na saída de um arquivo. Caso tudo tenha sido digitado manualmente, o texto tratado é exibido na tela, onde a *String* é simplesmente impressa através de uma chamada de sistema. Caso tenha sido aberta através de um arquivo, um novo arquivo com o nome "output" no formato TXT é criado, e a *String* armazenada dentro dele, tornando o processo de leitura e saída de arquivo automatizado. O programa então se encerra.

Algumas pseudo-instruções foram utilizadas na criação do programa. São elas:

- li $a1, 10 (move o valor 10 para o registrador 10)
- la $t1, texto (coloca o endereço de texto em $t1)
- beq $s1, $s2, label (jump para label se $s1 = $s2)
- bne $s1, $s2, label (jump para label se $s1 != $s2)
- move $s1, $s2 (move o conteúdo de $s2 em $s1)

3. Utilização

Inicie rodando o programa e algumas opções lhe serão apresentadas. Ao digitar a opção "1", será requisitado um nome de arquivo. Digite o nome juntamente com a extensão. Note que o arquivo deve estar na mesma pasta do executável do MARS, de preferência no formato TXT. Este arquivo deve conter um digito para conhecer o que deve ser feito com o texto ("1" para cifrar, "2" para decifrar), acrescido de um espaço, o fator da cifra a ser utilizado (Apenas dois dígitos, sendo o máximo 99 e o mínimo 01, com um "0" a frente do dígito quando menor que 10), novamente acrescido de um espaço, seguido pelo texto a ser tratado.

Exemplo: *1 03 A conduta dos jurados frente aos candidatos foi excelente.*

A mensagem será tratada e um arquivo com o nome "output.txt" será criado no mesmo diretório do arquivo de entrada.

Voltando ao início do programa, caso a opção "2" seja digitada, os dados deverão ser entrados manualmente. O segundo set de opções diz respeito ao que deve ser feito com o texto, digitando "1" para cifrar e "2" para decifrar. Em seguida, o programa gostaria de conhecer o texto a ser tratado, sendo necessário o usuário digita-lo e apertar ENTER. O valor do fator então será requisitado, bastando apenas o usuário entrar com o valor. Em seguida, o texto tratado deverá ser apresentado na tela e o programa finalizado.

4. Resultados e Troubleshooting

A seguir, um simples teste com diversos caracteres foi inserido no programa para comprovar seu funcionamento. Efetuado tanto através do processo de entrada manual quanto o de entrada via arquivo, os resultados apresentados foram os mesmos, comprovando a eficácia do software.

4.1 – Processo de Criptografia

```
----- BEM-VINDO | A CIFRA DE CÉSAR -----

Digite uma opção:
(1) Abrir texto e parâmetros do arquivo
(2) Digitar texto e parâmetros
OPÇÃO: 2

Digite uma opção:
(1) Criptografar
(2) Descriptografar
OPÇÃO: 1

Digite o texto: abc xwy ABC XWY 0123 789 "!@ [\~

Digite o valor do fator: 5

Resultado do texto: fgh cbd FGH CBD 5678 234 '&_ `{%


-- program is finished running --
```

4.2 – Processo de Descriptografia

```
----- BEM-VINDO | A CIFRA DE CÉSAR -----

Digite uma opção:
(1) Abrir texto e parâmetros do arquivo
(2) Digitar texto e parâmetros
OPÇÃO: 2

Digite uma opção:
(1) Criptografar
(2) Descriptografar
OPÇÃO: 2

Digite o texto: fgh cbd FGH CBD 5678 234 '&_ `{%

Digite o valor do fator: 5

Resultado do texto: abc xwy ABC XWY 0123 789 "!@ [\~
```

Todavia, o programa possui problemas de fácil resolução, mas que não foram aplicados em seu código. Para fins de estudo, iremos listar abaixo as deficiências e limitações mais comuns do software e propor uma solução.

- Nenhum dos *inputs* de opções de múltipla escolha possui tratamento caso o usuário digite um valor diferente dos propostos. Para evitar esse problema, bastava comparar se a opção digitada estava no intervalo proposto e, caso não, fazer com que o programa retorne ao *label* para digitar a opção, apresentando uma mensagem de "opção inválida".

- A opção de fator na leitura de arquivo trabalha com um máximo de 99, devido ao fato de o programa ler apenas dois dígitos após ler a opção de tratamento e antes de chegar até a mensagem. Uma possível solução para essa limitação é a de aumentar uma casa na leitura, acrescer um valor ao loop de verificação e multiplicar a casa excedente por 100 antes de somar com o resto, já que esta estaria entrando na casa das centenas.

5. Considerações Finais

De fácil compreensão e um desafio interessante para quem se interessa por computação e exatas, a Cifra de César é sem dúvidas um excelente objeto de estudo para quem possui interesse na área. Os obstáculos encontrados durante a confecção desse trabalho fizeram dele um método de aprendizado e aperfeiçoamento do conhecimento muito forte.

O software, que ainda apresenta deficiências e limitações, deve ser alvo de um empenho e dedicação maior para que esse seja concretizado e fique livre de bugs e falhas, porém ainda se mostra capaz de cumprir com a tarefa que lhe foi proposta com eficiência e clareza no seu funcionamento.

6. Referências

O Código de César, Disponível em: http://www.numaboa.com.br/criptografia/124-substituicao-simples/165-codigo-de-cesar

Criptografia, Disponível em: http://www.infowester.com/criptografia.php

Cartilha - Criptografia, Disponível em: http://cartilha.cert.br/criptografia/

MIPS Instruction Reference, Disponível em: http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html