

Project 1: ROT13

Due Thursday, September 23, 2004

Objective

This project is a finger-warming exercise to make sure that everyone can compile an assembly language program, run it through the debugger and submit the requisite files using the systems in place for the programming projects.

Background

The ROT13 format is used on USENET newsgroups to mask potentially offensive postings, movie spoilers, etc. The idea is that readers who think they might be offended by a controversial remark will simply not “decode” the posting and thus not be offended. Many news readers and email clients support ROT13.

The encoding is very simple. The characters ‘a’–‘m’ are mapped to ‘n’–‘z’ and vice versa. Upper case letters are transformed analogously. All other characters (e.g., digits and punctuation marks) are left alone. For example, “There was a man from Nantucket” becomes “Gurer jnf n zna sebz Anaghparg” after ROT13 transformation. To decode a message in ROT13, you simply apply the ROT13 transformation again.

Assignment

For this project, you must do the following:

1. Write an assembly language program that prompts the user for an input string and prints out the ROT13 encoding of the the string. A good starting point for your project is the program `toupper.asm` (shown in class) which converts lower case characters in the user’s input string to upper case. The source code is available on the GL file system at:

```
/afs/umbc.edu/users/c/h/chang/pub/cs313/
```

2. Using the UNIX `script` command, record some sample runs of your program and a debugging session using `gdb`. In this session, you should fully exercise the debugger. You must set several breakpoints, single step through some instructions, use the automatic display function and examine the contents of memory before and after processing. The `script` command is initiated by typing `script` at the UNIX prompt. This puts you in a new UNIX shell which records every character typed or printed to the screen. You exit from this shell by typing `exit` at the UNIX prompt. A file named `typescript` is placed in the current directory. You must exit from the `script` command *before* submitting your project. Also, remember not to record yourself editing your programs — this makes the `typescript` file very large.

Turning in your program

Use the UNIX `submit` command on the GL system to turn in your project. You should submit two files: 1) the modified assembly language program and 2) the `typescript` file of your debugging session. The class name for submit is `cs313_0101`, the project name is `proj1`. The UNIX command to do this should look something like:

```
submit cs313_0101 proj1 rot13.asm typescript
```

Notes

Additional help on running NASM, `gdb` and making system calls in Linux are available on the assembly language programming web page for this course:

```
<http://www.csee.umbc.edu/~chang/cs313.f04/assembly.shtml>
```

Recall that the project policy states that programming assignments must be the result of individual effort. *You are not allowed to work together.* Also, your projects will be graded on five criteria: correctness, design, style, documentation and efficiency. So, it is not sufficient to turn in programs that assemble and run. Assembly language programming can be a messy affair — neatness counts.