CSC 3410 Assignment 3

Testing Palindromes

Skills you will practice

Writing Loops (FOR and WHILE loops)
Writing Decision statements (IF statements)
Writing Functions

Things you will need

Virtualbox virtual machine (with all the tools)
A text editor (provide by the VM - VS code, Gedit, VIM, ...)
The Nasm assembler (provided by the VM)
The ld linker (provided by the VM)

Description

For this assignment, you are going to write an assembler program that will test if sentences are palindromes. The high-level pseudocode is as follows:

```
loop until the user is finished
    prompt the user for a string
    read a string from the keyboard
    test the string to determine if it is a palidrome
    if so, then
       print "It is a palindrome"
    else
        print "It is NOT a palindrome"
end loop
```

Your program will use the SYS_READ Linux call to read a sentence from the keyboard. The user will signify that they are finished by pressing <return> without entering a sentence. After a call to SYS_READ in which the user presses <return> without entering a string, the SYS_READ places a newline at the beginning of the buffer (a newline byte contains the value 10). So, you should test after the call to SYS_READ, and if the first byte in the buffer is a newline, then your program should exit nicely (by calling SYS_EXIT).

Also, you will implement a function called is_palindrome in assembler that your program will call to determine if a buffer contains a palindrome. The prototype for is_palindrome is as follows:

```
int is_palindrome(char *buffer, int len);
```

The function returns a 1 if the buffer contains a palindrome, or a 0 if the buffer does not contain a palindrome. Note that the len that is passed should be the length of the string not counting the newline at the end that SYS_READ includes. Also, remember that SYS_READ returns the number of bytes read in EAX (however, do not forget to decrement EAX by 1 so that you to not include the newline in the length). To get credit for this assignment, you must follow all C calling conventions for a 32 bit program on the Intel architecture (what we have been studying in class) for the is_palindrome function.

The program is to be written and tested on the virtual machine given in class, must compile using the nasm assembler, and must link using the ld linker. To compile the program, you will enter the following commands:

```
nasm -g -f elf -F dwarf -o palindrome.o palindrome.asm
ld palindrome.o -m elf_i386 -o palindrome
```

Do not use any macros or libraries for this assignment.

The following is a version of this program written in C that you can use as a reference. Your program should behave similarly to it.

```c
#include <stdio.h>
#include <unistd.h>

int is_palindrome(char * buf, int len);

int main() {
    char buf[1024];
    int count;
    write(1, "Please enter a string:\n", 23);
    count = read(0, buf, 1024);
    while (buf[0] != '\n') {
        count--;
        if (is_palindrome(buf, count)) {
            write(1, "It is a palindrome\n", 20);
        } else {
            write(1, "It is NOT a palindrome\n", 23);
        }
        write(1, "Please enter a string:\n", 23);
        count = read(0, buf, 1024);
    }
    return 0;

}

int is_palindrome(char *buf, int len) {
    int i, j;
    for (i = 0, j = len - 1; i < len/2; i++, j--)
        if (buf[i] != buf[j]) return 0;
    return 1;
}
```

The following shows an example run of this program:

```
$ ./pal_in_c
Please enter a string:
1221
It is a palindrome
Please enter a string:
rats live on no evil star
It is a palindrome
Please enter a string:
1
It is a palindrome
Please enter a string:
beef
It is NOT a palindrome
Please enter a string:
1122322211
It is NOT a palindrome
Please enter a string:
```

```
$
```

The following command will compile the above example program written in C:
`gcc -g -Wall -o pal_in_c pal_in_c.c`

(Note: if you have gcc produce assembler from the C code and submit the gcc generated assembler, then the professor will know and give you a grade of 0).

Turn In

Create a directory called <your_email>_hw3 and put your program in it, where <your_email> is your TnTech email id (**do not** including the @tntech.edu). Make sure you name the programs palindrome.asm. You will send your entire project's directory. You can create your tarball with the following command:
`tar -czf <your_email>_hw3.tar.gz <your_email>_hw3/`
from the top level directory (one up from your project directory). Submit the tarball via iLearn.

Due on Mar 31, 2023 11:59 PM
Available until Apr 3, 2023 11:59 PM. **Access restricted after availability ends.**

Attachments

CSC 3410 Assignment 3.docx (18.33 KB)

Download All Files