Project 4: C Functions

Due: Tue 10/14/03, Section 0101 (Chang) & Section 0301 (Macneil)
Wed 10/15/03, Section 0201 (Patel & Bourner)

Objective

The objective of this programming exercise is to practice writing assembly language programs that use the C function call conventions.

Assignment

Convert your assembly language program from Project 3 as follows:

1. Convert the program into one that follows the C function call convention, so it may be called from a C program. Your program should work with the following function prototype:

   ```
   void report (void *, unsigned int) ;
   ```

   The intention here is that the first parameter is a pointer to the records array and the second parameter has the number of items in that array.

2. Modify your program so it uses the strncmp() function from the C library to compare the nicknames of two records. The function prototype of `strncmp()` is:

   ```
   int strncmp(const char *s1, const char *s2, size_t n) ;
   ```

   The function returns an integer less than, equal to, or greater than zero if s1 (or the first n bytes thereof) is found, respectively, to be less than, to match, or be greater than s2.

3. Modify your program so that it prints out the entire record (not just the `realname` field) of the record with the least number of points and the record with the alphabetically first nickname. You must use the `printf()` function from the C library to produce this output. The output of your program would look something like:

   ```
   Lowest Points: James Pressman (jamieboy)
     Alignment: Lawful Neutral
     Role: Fighter
     Points: 57
     Level: 1
   First Nickname: Dan Gannett (danmeister)
     Alignment: True Neutral
     Role: Ranger
     Points: 7502
     Level: 3
   ```

A sample C program that should work with your assembly language implementation of the `report()` function is available on the GL file system: `/afs/umbc.edu/users/c/h/chang/pub/cs313/records2.c`

Implementation Notes

- Documentation for the printf() and strncmp() functions are available on the Unix system by typing `man -S 3 printf` and `man -S 3 strncmp`.
- Note that the strncmp() function takes 3 parameters, not 2. It is good programming practice to use `strncmp()` instead of `strcmp()` since this prevents runaway loops if the strings are not properly null terminated. The third argument should be 16, the length of the `nickname` field.

- As in Project 3, you must also make your own test cases. The example in `records2.c` does not fully exercise your program. As before, your program will be graded based upon other test cases. If you have good examples in Project 3, you can just reuse those.
- Use `gcc` to link and load your assembly language program with the C program. This way, `gcc` will call `ld` with the appropriate options:

```
    nasm -f elf report2.asm
    gcc records2.c report2.o
```

- Notes on the C function call conventions are available on the web:

```
    http://www.csee.umbc.edu/~chang/cs313.f03/stack.shtml
```

- Your program should be reasonably robust and report errors encountered (e.g., empty array) rather than crashing.

Turning in your program

Use the UNIX `submit` command on the GL system to turn in your project. You should submit at least 4 files: your assembly language program, at least 2 of your own test cases and a typescript file of sample runs of your program. The class name for submit is `cs313_0101`, `cs313_0102` or `cs313_0103` for respectively sections 0101 (Chang), 0201 (Patel & Bourner) or 0301 (Macneil). The name of the assignment name is `proj4`. The UNIX command to do this should look something like:

```
submit cs313_0103 proj4 report2.asm myrec1.c myrec2.c typescript
```