CIS 356 Computer Organization
Spring 2024

gStrLib — Using `strncat`

2024-04-16

Introduction

This is a group assignment on manipulating C-style (ASCIIZ) strings in MIPS assembly functions.

Assignment Goals

Learning Outcomes After completing this group assignment, each student is expected to be able to

- **Trace** C/Java code using `char *` parameters.
- **Translate** simple C/Java `char *` functions to MIPS.
- **Write** string functions in C/Java on the way to implementing them in MIPS.

Procedure

Get out paper for a *single* turn-in at the end of class. Copy enough of each question so that the paper could stand alone as a study guide.
Assign (Least-recently Held) Roles: *Manager*, *Recorder*, *Reflector*, *Speaker*. Everyone should help the whole team contribute and manage time.
Answer these questions:

1. Assume the call to the `new` **function** returns the *address* of the new object in `$v0` and that `SomeNode` has the following C/Java definition:

   ```
   class SomeNode {
     SomeNode * other;
     int radius;
     int height;
   };
   ```

   Translate the following C/Java to MIPS

   ```
   SomeNode * x = new SomeNode(); // Pretend no parameters for function
                                  // *you* save result in global x
   x->radius = 5;
   x->height = x->radius + 11;
   ```

2. Thinking about the `atoi` (ASCII-to-Int) code we wrote in class:
   (a) How did we convert from a `char` with an ASCII code of a digit to the numeric value of the digit?
   (b) Given a character variable, `ch`, that contains a *hexadecimal* digit (assume uppercase alphabetic), how could you use Java's `String.indexOf` function to convert to its numeric value?
   (c) Write `int hextoi(char * hexstring)` that converts the *non-null* string passed in into the equivalent `int` value.

3. Let's assume you have working C/Java versions of
   `int strlen(char * str)` and
   `char * strncpy(char * destination, char * source, int n)`
   (a) Write `char * chomp(char * str)` in C/Java **using** the given functions to make your code as simple as possible.
   (b) Write `char * strncat(char * destination, char * source, int n)` in C/Java **using** the given functions to make your code as simple as possible.
   (c) What value(s) in the *activation record* of `strncat` do you count on remaining unchanged on calling the other library functions?
   (d) In MIPS assembly what must you do with those values **before** calling the other function(s)?