Implementing C string libraries in SPARC V8 assembly language (with 64 bit extensions)

Gauri Patrikar[1] and Siddharth Rao Deb[1]

[1]Project Research Engineer, IITB

January 27, 2020

Abstract

The string library is a very commonly used library which provides a number of very useful functions like copying a string (strcpy) and comparing two strings (strcmp), and has been written in C. We will attempt to write these in Sparc v8 assembly with 64 bit extensions in order to reduce the number of instructions for execution of a task and hence increase speed. We will begin with a brief description of the string library and the 64 bit extensions used, and then give the final implementation.

1 Introduction

A string is basically an array of characters. The end of a string is given by the null character '/0' The string library contains functions that perform certain common operations on strings such as comparing, concatenating, copying pairs of string or computing the length of a string. We will also check for 64-bit alignment. The following functions are being implemented here -

1. strcpy
2. strncpy

3. strcmp

4. strncmp

1.1 A64strcpy

The A64strcpy() function is similar to the strcpy, it copies the string pointed to by the second pointer, including the terminating null byte, to the buffer pointed to by the first pointer. It can copy 64 bits in a single cycle. It begins by checking by checking for alignment. If it is 64 bit aligned it will execute the assembly code, if not it will execute the C code.

It conforms to the c99 standard (as it has the restrict qualifier).

1.1.1 Calling the function

```
 Include the following header in your program-

#include <A64string.h>

It can be called in the following format-

char *A64strcpy(char *dest, const char *src);


It returns the pointer to the destination
```

1.1.2 Implementation

```
    .    save %sp, -96,%sp
begin:  ld [%i0],%l0
        st %l0,[%i1]
        addd 0x40004,%i0
        zbytedpos %l0,0xff,%l1
        cmp %l1,0x0
        bne begin
        restore
        retl
        nop
```

1.1.3 Program Flow

- Load word from source memory
- store in dest. memory
- Check for null byte
- IF present, terminate, else repeat

2 Conclusion

Here an assembly implementation of the C string library was given. The language was sparc-v8 with its 64 bit extensions.