

Homework #4 – MIPS Assembly Programming II

CDA 3100, Computer Organization I, Fall 2007

Department of Computer Science, Florida State University

Points: 100 (6 % of the total grading)

Due: Beginning of class on October 10, 2007

Submission: Hardcopy required along with an electronic copy by electronic mail to teaching assistant Trung Nguyen at tdn07@fsu.edu.

Problem 1 (40 points) Write a MIPS program that will read in a base (as an integer) and a value (as an ASCII string) in that base and print out the decimal value. The base will be given in decimal and will be between 2 and 36, and the digits for the values will be in set $[0\dots9, a\dots z]$. You can assume that no overflow will occur. An example run will be:

Enter a base (between 2 and 36 in decimal): 2

Enter a number in base 2: 101

The value in decimal is: 5

Hint: The input and output can be done using syscall in MIPS that is given on p. A-44 (Appendix A).

Problem 2 (60 points) Write a MIPS program that will read in an image name (gray level image in pgm format), read in the image, compute its histogram, and then output the histogram. (A C version of the following and a main program is available at http://www.cs.fsu.edu/~liux/courses/cda3100/assignments/read_pgm.c, <http://www.cs.fsu.edu/~liux/courses/cda3100/assignments/histogram.c>, and http://www.cs.fsu.edu/~liux/courses/cda3100/assignments/hw6_prob2.c. Note that in MIPS, to allocate additional memory, use sbrk system call (system call number 9).

A pgm image starts with a required header on the first two lines. The first line is always “P5\n” and the second line consists of three numbers (in ASCII): the number of rows, the number of columns, and 255. The three numbers are separated by some white space(s), followed by a new line character (“\n”, which is 10 in decimal). For examples,

P5

128 256 255 (in memory, the hexadecimal values are “50350a31323820323536203235350a”)

is a header of an image with 128 rows and 256 columns. Following the header are the pixel values in binary and they are saved row by row and within each row from left to right; there is no white space between rows. Each pixel uses one byte (eight bits) as an unsigned char. The following C/C++ function reads a pgm image.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

unsigned char *read_pgm(char *fname, int *nrow, int *ncol)
{
    int fd;
    char str[256];
    char P5_str[32];
    int i, j, k;
    unsigned char *chp;
    unsigned char *image;

    fd = open(fname, O_RDONLY);
    if (fd < 0) {
        /* In the MIPS version, you can assume that the file is always valid */
        return (unsigned char *) NULL;
    }
}
```

```

read(fd, P5_str, (int)3);
/* Now we read the header of the pgm file */
k = -1;
do {
    k++;
    read(fd, &(str[k]), 1);
} while (str[k] != 10);
str[k] = 0;
sscanf(str, "%d%d", nrow, ncol);
printf("Rows %d columns: %d\n", (*nrow), (*ncol));
image = (unsigned char *)malloc((*nrow) * (*ncol));
chp = image;
for (i=0; i < (*nrow); i++) {
    for (j=0; j < (*ncol); j++) {
        read(fd, chp, 1);
        chp++;
    }
}
close(fd);
return image;
}

```

The histogram of an image is widely used to characterize the image in content-based retrieval and other applications. It counts the number of occurrences of all the pixel values (0 – 255 in this case). The following C/C++ function computes the histogram of the input image.

```

/* Here image is the starting address of image, which has nrow rows and
   ncol columns.
   h is the histogram of the image.
*/

int histogram(unsigned char *image, int nrow, int ncol, int *h)
{
    int i, j, k;
    unsigned char *p;

    for (i=0; i < 256; i++)
        h[i] = 0;
    p = image;
    k = 0;
    for (i=0; i < nrow; i++) {
        for (j=0; j < ncol; j++) {
            h[*p] ++;
            p++;
            k ++;
        }
    }
    return k;
}

```