

# CS360 Midterm Exam - March 23, 2011 - Jim Plank

Answer all questions. Don't write answers on this exam, please.

## Question 1

Write the `jassem` assembler for the following two procedures:

<pre>int y(int **a) {     return a[0][1]; }</pre>	<pre>char *read10() {     char *s;      s = (char *) malloc(10);     read(0, s, 10);     return s; }</pre>
---	--

## Question 2

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int recfind(char *s, int start)
{
    char *x;

    x = strchr(s+start, ' ');
    if (x == NULL) {
        return start;          /* HERE */
    }
    return recfind(s, (x-s)+1);
}

main(int argc, char **argv)
{
    int i;

    i = recfind(argv[1], 0);
    printf("%d\n", i);
}
```

The program [q2.c](#) is to the left.

Suppose we run this program as follows:

```
UNIX> q2 "Bombs Away"
```

The initial value of the frame pointer, when `main()` is first executed, is `0xffff4840`. The value of `argv` is `0xffff4860` and the value of `argv[1]` is `0xffff487c`. What I want you to do is to tell me the identity and value of *every* word on the stack that you can identify. If you know the identity, but not the value, state just the identity. You should include values that are *above and below* the current stack pointer.

## Question 3

Explain buffering in the C Standard I/O library: How is buffering implemented? Why is buffering implemented? Give an example of a program where buffering helps. Give an example of a program where buffering doesn't really help.

## Question 4

Write a program `findjim.c` that prints the relative pathname of all files reachable from the current directory that contain the substring "jim". Your program should only use one open file descriptor at a time.

## Question 5

Behold the program [q5.c](#):

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>

main(int argc, char **argv)
{
    int p, u, f, m;

    if (argc != 4) {
        fprintf(stderr, "usage: q4 p m u\n");
        exit(1);
    }

    p = O_WRONLY;
    if (strchr(argv[1], 'a') != NULL) p |= O_APPEND;
    if (strchr(argv[1], 'c') != NULL) p |= O_CREAT;
    if (strchr(argv[1], 't') != NULL) p |= O_TRUNC;
    if (strchr(argv[1], 'x') != NULL) p |= O_EXCL;

    sscanf(argv[2], "%o", &m);
    if (sscanf(argv[3], "%o", &u) == 1) umask(u);

    f = open("f1.txt", p, m);
    write(f, "XXX\n", 4);
    close(f);

    system("ls -l f1.txt");
    chmod("f1.txt", 0644);
    system("cat f1.txt");
    exit(0);
}
```

I am going to give you twelve scenarios in which you execute two commands from the command line. The first does "**ls -l f1.txt**". The second calls **q5** with a variety of command line arguments. Your job is to tell me the output of the **q5** call. Don't bother differentiating standard output and standard error and don't worry about the dates in the files. Just to be clear, **ls -l** prints:

*mode number-of-links owner-username owner-groupname size modification-date filename*

If the program has a segmentation violation, let me know that. In all cases, if the file **f1.txt** exists, it contains the seven bytes: "**Shaft!\n**". For this problem, your username is **plank**, and your group name is **staff**.

The scenarios are on the answer sheet -- put your answers there.