Homework 4

Due Date: Monday, May 27, 2002, ***at 11:59PM*** **Points**: 100

UNIX System

1. (*10 points*) How do I delete the file –i?

2. (*15 points*) Please list all sections of the manual on the CSIF Linux systems containing a command named *time*. What are the file names of those manual pages?

C Programming

3. (*30 points*) A *palindrome* is a word or sentence which is the same whether read backwards or forwards. Please write a program ispal.c which reads each line from its standard input and says whether or not the line is a palindrome. For each line, all non-alphanumeric characters are to be ignored, and case is not to matter. For example,

standard input	***standard output***
`Madam, I'm Adam!`	`A palindrome`
`O give me a home`	`Not a palindrome`
`Able was I ere I saw Elba`	`A palindrome`

Hint: use recursion.

4. (*30 points*) Write a program to list the files in a directory in the order in which the directory entries occur. Your program is to use the library functions *opendir* (3), *readdir*(3), and *closedir*(3). Your output should include the number of the directory entry, the inode number of the file, and the file name. For example, in a directory containing the files a.out and d.c, the output might look like:

```
order     inode         name
    1      1119713      .
    2       272679      ..
    3      1119719      d.c
    4      1119738      a.out
```

Hint: On the Linux systems in the CSIF, the inode field of the dirent structure is d_ino. Please see the manual pages for the functions for more details. (The inode field is *not* documented there, but the field containing the name of the file is.)

Debugging

5. (*15 points*) On the web page for the class is the source for a set of routines to manage a linked list. The functions provided are:

```
int insert(struct link **head, int value)
```

Insert value into the linked list pointed to by *head; update *head if necessary. If the insertion succeeds, return 1; if it fails, return 0.

```
int delete(struct link **head, int value)
```

Delete value from the linked list pointed to by *head; update *head if necessary (if the list is empty, make it NULL). If the deletion succeeds, return 1; if it fails (because value is not in the list), return 0.

```
void prforw(struct link *head)
```

Print all integers in the linked list pointed to by head in order (from head to tail).

```
void prback(struct link *head)
```

Print all integers in the linked list pointed to by head in reverse order (from tail to head).

Unfortunately, they don't quite work right. Debug them. Turn in commented, corrected routines; be sure you

mark what changes were made, and why!

Hints: The program testll.c is a program that calls the library routines to manage a linked list of numbers. You can use it to help figure out what the problems are. The manual page on the web tells you how to use that program. Both the file containing the routines, llist.c, and the test program, testll.c, require the header file llist.h.

Extra Credit

6. (*10 points*) Modify the program you wrote in problem 4 to use the *scandir*(3) function to sort the files by either inode number (option –i) or name (option –n).