sorter state and data str. design.

ACM Student Chapter Meeting Topic: Lecture 14

CSI 310: Lecture 14
and I in arranging for meeting areas and in planning of events.
The heads of each one of the SIGs will work with me
head for each SIG. The heads of each SIG will part off into groups and select
stay who are interested and we will develop SIG to work on our web page. At the end of the meeting
6. Formation of a Programming Competition SIG as well as formation of a

problem and solution,
5. ACM Programming Competition. A handout will be given out with a
out on common myths of BSD.
4. Presentation from Andy Katz on Linux vs. BSD. A handout will be

AGENDA FOR THE MEETING WILL BE AS FOLLOWS...

Subject: [VALBANY-ACM] Reminder: Meeting Thursday (3/18) @ 7:15 IC224
To: ACM@isterv.albany.edu
From: Justin Litz <152560@valbany.edu>
Date: Wed., 17 Mar 2004 20:33:09 -0500
University at Albany Computer Science Dept.
Justin Lutz

I will try to get food for the meeting but I cannot guarantee it.

If anyone has any questions or cannot make the meeting let me know.
State Diagram for sorter

Inputs
- Char buffer [BUFSIZE]

Data Structure Diagram

Commands
- Read and Dispatch
- Merge sort
- Print split
- Print forward
- Split

Command Phase
- Lines
- Process Text
- Exit

Input Phase
- Start

Read and Process Text

Some ot 34 text 0
Here are sw
These are sw

NULL
NULL

HEAD
HEAD2

TAIL
TAIL

Char buffer [BUFSIZE]
return 0;

while (!finished) //
    // cout << A >> endl; // A sorted string was printed.
    now, A[0..nch-1] is sorted

    now, A[i] has the smallest char from A[i..nch-1]
    {
        // code to "swap" A[i], A[j]
        if (A[i] < A[j])
            for (j = i+1; j < nch; j++)
                for (i = 0; i < nch-1; i++)
                    // A[nch] = string A

        while (cin >> string(A, A.size()) >> A[i], i = 0, A[j], j)
            char A[i size];
    }

    cin.getline(A, 100); int nch; // char A[nch];
    using namespace std; main()

    #include <iostream>
    #include <string>

    Selection sort demo: processes chars within the array A.
For next week, after the midterm, we will discuss the (recursive) MergeSort algorithm, a topic in the other half of the course.

We now illustrate what half your Project 3 work must do.

C-strings.

Sort the names of landmarks in Project 2 "main list" lexicographically, as this program manipulates characters as if they were numbers!
How can we very efficiently swap the strings in the nodes pointed to by $i$ and $j$?
Computer does not copy chars or node pointers!

Swap the values in the 2 data fields of the nodes pointed to by I and J.

pT = I->data; I->data = J->data; J->data = pT;

NULL

I

TONE

two

THREE

TAIL

HEAD

char *
pT;
And now for a new show...