COP4530 – Data Structures, Algorithms and Generic Programming
Recitation 5
Date: 30th September, 2011

Lab topic:

1) **Discuss Assignment 3**
2) **Take Quiz 4**

<u>Self Organizing link lists:</u>

Searching for an element in a doubly link list takes O(n) time where n represents the number of elements in the list. Using self organizing link list we can get a better expected running time.

In Assignment 3 you have to store the words in **<u>two separate</u>** self organizing doubly linked lists (One for the standard dictionary words found at /usr/share/dict/words and the other for the user dictionary). The self organizing property is defined as follows:

- With each word there will be a frequency field that denotes number of times it was encountered while parsing the current input file. This field is initialized to 0 and is increased by 1 each time.
- Whenever the value of this field is equal to 4 for a certain word, that word should be moved to the front of the list.

Finally you have to output the first three words in the self-organizing linked list for the standard dictionary before quitting.

<u>Average Case Analyses for Self Organizing link lists:</u>

In general say a self organizing list contains *n* elements and probability for searching for the *i*th element is denoted by p[i] (should be read as p subscript I). What is the average search time for that list ?

[Example covered in class are all p[i] = 1/n and all p[i] = 1/(2*n)]

<u>Using `gettimeofday` to time portions of code</u>

For Assignment 3 you also have to output the time taken, in seconds, to store the contents of the standard dictionary in the list. For that purpose you can use the function

Here is the prototype for the function:

```
int gettimeofday(struct timeval *tp, NULL)
```

The first parameter must be a pointer to a previously declared timeval variable (or in C, a struct timeval variable). A timeval has two components, both ints. One (called tv_sec) is the time in seconds since 1/1/1970 (epoch).The other (called tv_usec) is the number of microseconds into that second. Example:

```
timeval tim;
gettimeofday(&tim, NULL);
double t1=tim.tv_sec+(tim.tv_usec/1000000.0);
do_something_long();
gettimeofday(&tim, NULL);
double t2=tim.tv_sec+(tim.tv_usec/1000000.0);
printf("%.6lf seconds elapsed\n", t2-t1);
```

The second parameter (NULL) used to be to retrieve the local time zone, but time zones are no-longer handled that way. [Ref: http://rabbit.eng.miami.edu/info/functions/time.html#gtod]

A sample code that outputs the time spent in iterations:

```
#include <time.h>
#include <sys/time.h>
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char **argv)
{
  if (argc < 2)
    {
      printf("USAGE: %s loop-iterations\n", argv[0]);
      return 1;
    }

  int iterations = atoi(argv[1]);

  struct timeval start, end;
  gettimeofday(&start, NULL);

  for (int i = 0; i < iterations; i++)
    {
    }

  gettimeofday(&end, NULL);

  printf("%ld\n", ((end.tv_sec * 1000000 + end.tv_usec)
                  - (start.tv_sec * 1000000 + start.tv_usec)));
  return 0;
}
```

[Code Ref: http://www.cs.loyola.edu/~jglenn/702/S2008/Projects/P3/time.html]