Computer Science 1510

Lecture 32

March 30, 2016

Lecture Outline

- Programming examples

Final exam:
Wednesday, April 13
12 – 2 pm
EN-1001

Example 1: Selection sort (integers)

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

void swap(int *a, int *b);

int main(int argc, char *argv[])
{
   int i,j,n,min,alloc;
   int *sorted;
   int *temp;
   char buff[256];
   FILE *fpin, *fpout;

   fpin=fopen("input_list.dat","r");    /* Input file */
   if (fpin==NULL){
      printf("Unable to open input file\n");
      return -1;
   }
   fpout=fopen("output_list.dat","w");  /* Output file */
   if (fpout==NULL){
      printf("Unable to open output file\n");
      return -1;
   }

   i=0;
   sorted=(int*)calloc(10,sizeof(int));
   alloc=10;

   /* Read in the integers */
   while (fgets(buff,256,fpin)!=NULL){
      if (i>alloc-1){  /* Not enough space, allocate more memory */
         temp=(int*)realloc(sorted,(alloc+10)*sizeof(int));
         if (temp!=NULL) sorted=temp;
         alloc+=10;
      }
```

```
      sscanf(buff,"%d",&sorted[i]);
      i++;
   }
   n=i;
   fclose(fpin);

   /* Use selection sort to sort the integers */
   for (i=0;i<n-1;i++){
      min=i;
      for (j=i+1;j<n;j++){
         /* Determine the minimum of the elements from i to n */
         if (sorted[j]<sorted[min]) min=j;
      }
      /* Swap minimum value with value in position i */
      swap(&sorted[i],&sorted[min]);
   }
   for (i=0;i<n;i++){
      fprintf(fpout,"%d\n",sorted[i]);
   }
   fclose(fpout);
   free(sorted);

   return 0;
}

void swap(int *a, int *b){
   int temp;
   temp=*a;
   *a=*b;
   *b=temp;
   return;
}
```

Example 2: Selection sort (strings)

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

void swap(char *array[], int a, int b);

int main(int argc, char *argv[])
{
   int i,j,min;
   const int n=10;
   char *names[n];  /* n pointers to characters */
   FILE *fpin, *fpout;

   fpin=fopen("classlist.dat","r");    /* Input file */
   if (fpin==NULL){
      printf("Unable to open input file\n");
      return -1;
   }
   fpout=fopen("alphabetic.dat","w");  /* Output file */
   if (fpout==NULL){
      printf("Unable to open output file\n");
      return -1;
   }

   /* Read in the names */
   for (i=0;i<n;i++){
      names[i]=(char*)malloc(256);
      fscanf(fpin,"%s",names[i]);
   }
   fclose(fpin);

   /* Use selection sort to sort the names */
   for (i=0;i<n-1;i++){
      min=i;
      for (j=i+1;j<n;j++){
```

```
         /* Determine the minimum of the elements from i to n */
         if (strcmp(names[j],names[min])<0) min=j;
      }
      swap(names,i,min);
   }
   for (i=0;i<n;i++){
      fprintf(fpout,"%s\n",names[i]);
      free(names[i]);
   }
   fclose(fpout);

   return 0;
}

void swap(char *array[], int a, int b){
   char *temp;
   temp=array[a];
   array[a]=array[b];
   array[b]=temp;
   return;
}
```

Example 3: Selection sort (strings)

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

void swap(char **array, int a, int b);

int main(int argc, char *argv[]) {
   int i,j,min;
   int n=0; /* number of names. */
   int size=1; /* current number of slots allocated */
   char **names=NULL;
   char tmp[1024]; /* tmp buffer to hold one line */
   FILE *fpin, *fpout;

   fpin=fopen("classlist.dat","r");    /* Input file */
   if (fpin==NULL){
      printf("Unable to open input file\n");
      return -1;
   }
   fpout=fopen("alphabetic.dat","w");  /* Output file */
   if (fpout==NULL){
      printf("Unable to open output file\n");
      return -1;
   }

   /* allocate space for one name */
   names=(char**)calloc(1,sizeof(char*));

   /* read names as long as there is data */
   while (fgets(tmp,1024,fpin)!=NULL) {
      n++;
      if (n>size) { /* then we must allocate more slots */
         size*=2;
         char **p=realloc(names,size*sizeof(char*));
         if (p==NULL) {
```

```
            printf("Ran out of memory! Size=%d\n",size);
            return -1;
         }
         names=p;
      }
      names[n-1]=(char*)malloc(256);
      sscanf(tmp,"%256s",names[n-1]);
   }

   printf("There were %d names read from the file.\n",n);
   fclose(fpin);

   /* Use selection sort to sort the names */
   for (i=0;i<n-1;i++){
      min=i;
      for (j=i+1;j<n;j++){
         /* Determine the minimum of the elements from i to n */
         if (strcmp(names[j],names[min])<0) min=j;
      }
      swap(names,i,min);
   }
   for (i=0;i<n;i++){
      fprintf(fpout,"%s\n",names[i]);
      free(names[i]);
   }
   free(names);
   fclose(fpout);

   return 0;
}

void swap(char **array, int a, int b){
   char *temp;
   temp=array[a];
   array[a]=array[b];
   array[b]=temp;
   return;
}
```