Assignment = Homework#03(CIS340)
Name/ID# = Daniel Izadnegahdar / 2420596
Name/ID# = Kendra Dennis / 2637975

CLEVELAND STATE UNIVERSITY

1. ***Problem#01:***
   1.1. ***Question:*** *Write a program that reads a text file and prints out any words that begin with a user-given string. The filename should be given at the command line as an argument. The program should prompt the user for the search string. The program should then read the file one word at a time and print out the word if its first N bytes match the search string. Where N is the length of the search string.*
   1.2. ***Answer:***
      1.2.1. ***Code:*** **Provided as *hw03-01.c* & *List.txt***

```c
//Import libraries
  #include <stdio.h> //This was added for standard i/o imports
  #include <string.h> //This was added to use strcmp and memset for
//Main function
  int main(int argc, char *argv[])
    {//Start of main
       //Variables
         char key[100];
         int keyLength = 0;
         int charMatch = 0;
         int i, j, k;
       //Begin program
         printf("****************hw03-01.c****************\n");
       //Check that 2 arguments are provided
         if (argc != 2)
           {//Start of if
              printf("ERROR! You didn't enter the 2nd argument for the file name!\n");
              printf("CORRECT EXAMPLE: ./a.out list.txt\n");
              return 0;
           }//End of if
       //Get the user key + keyLength
         printf("Enter the key to search: ");
         scanf("%s", key);
         keyLength = strlen(key);
       //Scan each line of the document
         //Variables
           int masterStringMaximum = 1000;
           char masterString[masterStringMaximum];
           char* filename = argv[1];
           FILE *filePointer;
           filePointer = fopen(filename, "r");
         //If error opening file, notify and exit
            if (filePointer == NULL)
              {//start of if
                printf("the file could not be opened!\n");
                printf("did you type the file name correctly including the extension?\n");
                printf("CORRECT EXAMPLE: ./a.out list.txt\n");
                return 0;
              }//end of if
         //If success opening file, scan each line and print if there's a match up to N
            while (fgets(masterString, masterStringMaximum, filePointer) != NULL)
              {//Start of while
                charMatch = 0;
                for(i = 0; i < keyLength; i++)
                  {//Start of for
                    if(masterString[i] == key[i])
                      {charMatch++;}
                  }//End of for
                if(charMatch == keyLength)
                  {{printf("%s", masterString);}}
              }//End of while
           fclose(filePointer);
           return 1;
  }//End of main
```

1.2.2. ***List.txt content:***

he
he
hell
hell
hello
hello

1.2.3. ***Output:***

```
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-01$ gcc hw03-01.c
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-01$ ./a.out list.txt
*****************hw03-01.c*****************
Enter the key to search: he
he
he
hell
hell
hello
hello
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-01$ ./a.out list.txt
*****************hw03-01.c*****************
Enter the key to search: hell
hell
hell
hello
hello
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-01$ ./a.out list.txt
*****************hw03-01.c*****************
Enter the key to search: hello
hello
hello
```

---

2. ***Problem#02:***

2.1. ***Question***:

Write out the memory map for the following code, providing all values at the end of execution.

```
#include <stdio.h>

typedef struct amount {
                        int dollars;
                        char cents;
                        } money;


main()
{
money a,*b;

a.dollars=1;
a.cents=99;
b=&a;
b->cents=75;
}
```

*2.2. **Answer**:*

Label	Address	Value
a.dollars	400 - 403	1
a.cents	404 - 405	~~99~~ 75
b	406 - 409	400

---

3. *Problem#03:*

3.1. Question:

Write a program to list all files and sub-directories in a current directory.

3.2. Answer:

3.2.1. Code: **Provided as *hw03-03.c***

```
1 //This program will list all the files and sub-directories of a directory.
2 //It can also show specific files if a 2nd argument is provided.
3 //The syntax used is similar to the ls1.c program of deck#08a/slide#11.
4 //Libraries
5   #include <dirent.h> //Used for the dirent struct
6   #include <stdio.h> //Used for standard I/Os
7 //Function declarations
8   void showContent(char []);
9 //Main function
10  int main(int argc, char *argv[])
11    {//Start of main
12       //Notify start
13         printf("**********************hw03-03.c**********************\n");
14       //If no addtional arg is provided, show all...
15         if(argc == 1)
16           {showContent(".");}
17         else
18       //Else if an additional arg is provided, filter accordingly...
19       {//Start of else
20         while(--argc)
21           {//Start of while
22             printf("%s\n", *++argv);
23           }//End of while
24       }//End of else
25    }//End of main
26 //ShowContent function
27  void showContent(char directoryLocation[])
28    {//Start of showContent
29      DIR *directoryPointer; //main directory
30      struct dirent *direntPointer; //directory entries
31      if((directoryPointer = opendir(directoryLocation)) != NULL)
32        {//Start of if
33          while((direntPointer = readdir(directoryPointer)) != NULL)
34            {//Start of while
35               if((*direntPointer).d_type == DT_REG) //Show files
36                 {printf("[file] %s\n", (*direntPointer).d_name);}
37               else //Show directories
38                 {printf("[directory] %s\n", (*direntPointer).d_name);}
39            }//End of while
40              closedir(directoryPointer);
41        }//End of else
42    }//End of showContent
```

3.2.2. Output:

```
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-03$ gcc hw03-03.c
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-03$ ./a.out *
**********hw03-03.c**********
a.out
hw03-03.c
testDirectory01
testDirectory02
testFile01.txt
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-03$ ./a.out t*
**********hw03-03.c**********
testDirectory01
testDirectory02
testFile01.txt
```

3.2.3. Directory Content:

---

4. Problem#04:

4.1. Question:

What does the standard cp do if you try to copy a file onto itself? For example: cp file1 file1. What do you think is the correct action? Modify cp1.c to handle that situation.

4.2. Answer:

4.2.1. Word answer:

If a file file01.txt is in a folder, and the following command is executed in Linux:

cp file01.txt file.txt

it will not create a copy, and will provide this message:

cp: cannot stat 'file01.txt' and 'file01.txt' are the same file

The correct action would be to create a copy of the file and paste it with a slightly different name, like file01[new].txt.

4.2.2. Code:

Provided as *hw03-04.c*

```
1 //This is a modified version of cp1.c from deck#07b/slide#08
2 //The modification is that when the source and destination files are the same, it will
  concatenate the destination file with [new], and new&paste a separate file.
3
4 //Libraries
5   #include <stdlib.h>
6   #include <stdio.h>
7   #include <unistd.h>
8   #include <fcntl.h>
9   #include <string.h> //Added for the modification
10
11 //Variables
12   #define BUFFERSIZE 4096
13   #define COPYMODE 0644
14
15 //Function delarations
16   void oops(char *, char *);
17
18 //Main function
19   int main(int ac, char *av[])
20     {//Start of main
21        //Variables
22          int in_fd, out_fd, n_chars;
23          char buf[BUFFERSIZE];
```

```c
//**********************modification start*****************************
        printf("*********hw03-04.c*********\n");
        //Check to see if the 2 fileNames are the same.
        //If they are the same, add (new) to the destination name
          int i, j, k;
          int breakFileName = 0;
          char fileName[100];
          char fileExtension[100];
        //If the source and destination file is the same...
          if(strcmp(av[1], av[2]) == 0)
            {//start of if
             for (i = 0; i < strlen(av[2]); i++)
               {//Start of for
                  //Once it finds the . separator, split name and extension
                    if(av[2][i] == '.')
                      {breakFileName = 1;}
                  //Store the fileName
                    if (breakFileName == 0)
                      {fileName[i] = av[2][i];}
                  //Store the fileExtension
                    else if (breakFileName == 1)
                      {fileExtension[i - strlen(fileName)] = av[2][i];}
                }//End of for
            //Close the name and extension strings with '\0'
              fileName[i] = '\0';
              fileExtension[i - strlen(fileName)] = '\0';
            //Concatenate the destination file with [new]
              strcat(fileName, "[new]");
              strcat(fileName, fileExtension);
              strcpy(av[2], fileName);
           }//end of if
//**********************modification end*****************************
       //Check that there are 3 arguments
         if( ac != 3 )
           {//Start of if
             fprintf( stderr, "usage: %s source destination\n", *av);
             exit(1);
           }//End of if
       //Check that files can be opened
         if((in_fd = open(av[1], O_RDONLY)) == -1)
           {oops("Cannot open ", av[1]);}
       //Check that files can be created
         if((out_fd = creat(av[2], COPYMODE)) == -1)
           {oops( "Cannot creat", av[2]);}
       //If the arg, open, and create checks passed, new files
       //Read from source to buffer and transfer to destination
         while((n_chars = read(in_fd , buf, BUFFERSIZE)) > 0)
           {//Start of while
             if(write(out_fd, buf, n_chars) != n_chars)
               {oops("Write error to ", av[2]);}
           }//End of while
       //Check for file read
         if(n_chars == -1)
           {oops("Read error from ", av[1]);}
       //Check for file close
         if(close(in_fd) == -1 || close(out_fd) == -1)
           {oops("Error closing files","");}
    }//End of main
//Oops function
  void oops(char *s1, char *s2)
    {//Start of oops
      fprintf(stderr,"Error: %s ", s1);
      perror(s2);
      exit(1);
    }//End of oops
```

4.2.3. Output:

```
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-04$ gcc hw03-04.c
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-04$ ls
a.out  file01.txt  hw03-04.c  restore
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-04$ ./a.out file01.txt file01.txt
*********hw03-04.c*********
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-04$ ls
a.out  file01[new].txt  file01.txt  hw03-04.c  restore
```

.....

5. Problem#05:

5.1. Question:

The who1 program lists every entry in the utmp file. That was not our intention but it provided a handy tool for examining the contents of utmp. Make who1 more useful by adding code to it that prints out all the other fields in the struct. The ut_type filed is particularly useful.

5.2. Answer:

5.2.1. Code:

Provided as *hw03-05.c*

```c
//Libraries
  #include <stdio.h>
  #include <utmp.h>
  #include <fcntl.h>
  #include <unistd.h>
  #include <stdlib.h>
  #define SHOWHOST

//Prototype
  void show_info(struct utmp *utbufp);

//Main function
  int main()
    {//Start of main function
       //Variables
         struct utmp current_record;
         int utmpfd;
         int reclen = sizeof(current_record);
       //Introduce
         printf("******************hw03-05.c************\n");
       //Open the utmp_file
         if((utmpfd = open(UTMP_FILE, O_RDONLY)) == -1)
           {//Start of if
             perror( UTMP_FILE );
             exit(1);
           }//End of if
         while(read(utmpfd, &current_record, reclen) == reclen)
           {//Start of while
           show_info(&current_record);
           }//End of while
        //Close the file
          close(utmpfd);
          return 0;
    }//End of main function

//Show_info function
  void show_info(struct utmp *utbufp)
    {//Start of show_info
       printf("%-8.8s", utbufp->ut_name);//Login or user name
       printf(" ");//Space
       printf("%-8.8s", utbufp->ut_line);//Device name
       printf(" ");//Space
       printf("%4d", utbufp->ut_time);//Login time
       printf(" ");//Space
     //Added struct properties
       printf("%4d", utbufp->ut_type);//Entry type
       printf(" ");//Space
       printf("%4s", utbufp->ut_id);//Id
       printf(" ");//Space
       printf("%6d", utbufp->ut_pid);//Process id
       printf("   ");//Space
     #ifdef SHOWHOST
       printf("(%s)", utbufp->ut_host);//Host name
     #endif
       printf("\n");
    }//End of show_info
```

5.2.2. Output:

```
monet:~/Desktop/CIS340/hw03/hw03-05% gcc who.c
monet:~/Desktop/CIS340/hw03/hw03-05% ./a.out
reboot   ~         1509061745 (4.4.0-97-generic)
runlevel ~         1509061745 (4.4.0-97-generic)
LOGIN    tty4      1509061746 ()
LOGIN    tty5      1509061746 ()
LOGIN    tty2      1509061746 ()
LOGIN    tty3      1509061746 ()
LOGIN    tty6      1509061746 ()
LOGIN    tty1      1509061756 ()
daizadne :0        1509120921 (:0)
amchaudh pts/0     1509062069 ()
daizadne pts/2     1509120973 (:0)
monet:~/Desktop/CIS340/hw03/hw03-05% gcc hw03-05.c
monet:~/Desktop/CIS340/hw03/hw03-05% ./a.out
******************hw03-05.c******************
reboot   ~         1509061745    2   ~~      0   (4.4.0-97-generic)
runlevel ~         1509061745    1   ~~     50   (4.4.0-97-generic)
LOGIN    tty4      1509061746    6    4   1116   ()
LOGIN    tty5      1509061746    6    5   1120   ()
LOGIN    tty2      1509061746    6    2   1132   ()
LOGIN    tty3      1509061746    6    3   1133   ()
LOGIN    tty6      1509061746    6    6   1137   ()
LOGIN    tty1      1509061756    6    1   1867   ()
daizadne :0        1509120921    7   :0   5617   (:0)
amchaudh pts/0     1509062069    8   /0      0   ()
daizadne pts/2     1509120973    7   /2   6310   (:0)
```

added

---

6. Problem#06:

6.1. Question:

6.1.1. *The following code prints the permission as number. Modify the code to print the 10 lowest significant binary bits (e.g., "0110100100") for permission.*

```
#include <stdio.h>
#include <sys/stat.h>  /* needed for stat() function */

int main(int argc, char *argv[])
{
struct stat     fileinfo;  /* returned info about file */
int             i;

if (argc != 2)
  {
  printf("Usage:  statfile filename\n");
  exit(0);
  i=stat(argv[1],&fileinfo);
  if (i == -1)
    {
    printf("Unable to stat %s\n",argv[1]);
    exit(0);
    }
  printf("size: %d\n",fileinfo.st_size);
  printf("permissions: %d\n",fileinfo.st_mode);
  printf("last modified: %d\n",fileinfo.st_mtime);
  }
```

6.2. Answer:

6.2.1. Code:

Provided as *hw03-06.c*

```c
//Libraries
  #include <stdio.h>
  #include <stdlib.h>
  #include <sys/stat.h>
  #include <string.h>

//Prototypes
  char convertBit01(int octalEntry);
  char convertBit02(int octalEntry);
  char convertBit03(int octalEntry);

//Main function
  int main(int argc, char *argv[])
    {//Start main
       //Variables
         struct stat fileinfo;
         int i, j, k;
       //If there aren't 2 arguments, error out
         if(argc !=2)
           {//Start of if
             printf("Usage:   statfile filename\n");
             exit(0);
           }//End of if
       //Get the info of the file
         i=stat(argv[1], &fileinfo);
         if(i==-1)
           {//Start of if
             printf("Unable to stat %s\n",argv[1]);
             exit(0);
           }//End of if
       //Print the permission bits
           int modeDigit = fileinfo.st_mode;
       //Convert to octal
         //Variables
           int octalCount = 8;
           int modeOctal = 0;
           int remainder = 0;
           int offset = 1;
           int increment = 10;
           i = offset;
         //Through the tokens to convert
           while(fileinfo.st_mode != 0)
             {//Start of while
               remainder = fileinfo.st_mode % octalCount;
               fileinfo.st_mode = fileinfo.st_mode / octalCount;
               modeOctal += (i * remainder);
               i *= increment;
             }//End of while
       //Convert to binary
         //Variables
           char octal[100];
           char binary[100];
         //Convert int modeOctal to string octal
           sprintf(octal, "%d", modeOctal);
         //Convert the octal to the 3-bit sets
           binary[0] = octal[2];
           binary[1] = convertBit01(octal[3]);
           binary[2] = convertBit02(octal[3]);
           binary[3] = convertBit03(octal[3]);
           binary[4] = convertBit01(octal[4]);
           binary[5] = convertBit02(octal[4]);
           binary[6] = convertBit03(octal[4]);
           binary[7] = convertBit01(octal[5]);
           binary[8] = convertBit02(octal[5]);
           binary[9] = convertBit03(octal[5]);
           binary[10] = '\0';
       //Print the values
         printf("**************hw03-06.c**************\n");
         printf("Permissions(digit format): %d\n", modeDigit);
         printf("Permissions(octal format): %d\n", modeOctal);
         printf("Permissions(10-bit format): %s\n", binary);
    }//End of main

//Convert 1st char
  char convertBit01(int octalEntr
    {//Start of convertBit01
      switch(octalEntry)
        {//Start of switch
          case '0': return '0';
          case '1': return '0';
          case '2': return '0';
          case '3': return '0';
          case '4': return '1';
          case '5': return '1';
          case '6': return '1';
          case '7': return '1';
          default: break;
        }//End of switch
    }//Start of convertBit01

//Convert 2nd char
  char convertBit02(int octalEntry)
    {//Start of convertBit02
      switch(octalEntry)
        {//Start of switch
          case '0': return '0';
          case '1': return '0';
          case '2': return '1';
          case '3': return '1';
          case '4': return '0';
          case '5': return '0';
          case '6': return '1';
          case '7': return '1';
          default: break;
        }//End of switch
    }//Start of convertBit02

//Convert 3rd char
  char convertBit03(int octalEntry)
    {//Start of convertBit03
      switch(octalEntry)
        {//Start of switch
          case '0': return '0';
          case '1': return '1';
          case '2': return '0';
          case '3': return '1';
          case '4': return '0';
          case '5': return '1';
          case '6': return '0';
          case '7': return '1';
          default: break;
        }//End of switch
    }//End of convertBit03
```

6.2.2. ***Output:***

```
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-06$ gcc hw03-06.c
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-06$ ./a.out test.txt
**************hw03-06.c**************
Permissions(digit format): 33188
Permissions(octal format): 100644
Permissions(10-bit format): 0110100100
dizad@dizad-HP-EliteBook-8560p:~/Desktop/cis340/hw03/hw03-06$ ./a.out a.out
**************hw03-06.c**************
Permissions(digit format): 33277
Permissions(octal format): 100775
Permissions(10-bit format): 0111111101
```