COSC 350 System Software (Mini Test #2)

10/04/21

Name: _____

1.

- Contiguous allocation – each file is saved in contiguous block. Simple and quick seek time. File name, the first block address and number of blocks are saved in the directory.
- Linked-list allocation – each block is used save data and the next block address. File name and the first block address for each file are saved in the directory.
- Linked-list allocation with FAT(file allocation table) – entire file information are saved in FAT. FAT must be loaded in RAM. wasting memory space
- Index-Node – each file has a corresponding i-node. Each file information (blocks address used) are saved in i-node. Only i-nodes currently opened files need be loaded in RAM. File name and i-node number are saved in the directory

2.

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

int main(int argc, char *argv[])
{
    int offset = 0;
    char a;
    int fd = open(argv[1],O_RDONLY);
    while(read(fd, &a, 1) == 1)
          offset++;
    printf("size of \"%s\" is %d bytes\n",argv[1], offset);
    close(fd);
    exit(0);
}
```

3.

```
#include <stdio.h>
#include <stdlib.h>
int st_to_int(char *);
void main(int argc, char *argv[])
{
     int i, num;
     int esum =0;
     int osum =0;
     if (argc <= 1)// argment must be at least two or more
     {
          printf("argument number error \n");
          exit(1);
     }

     for (i=1; i<argc; i++)//read command line input
     {
          num = st_to_int(argv[i]);
          if ((num % 2)== 0)
               esum = esum + num;
          else
               osum =osum+ num;
     }

     printf("The sum of even arguments is %d\n", esum);
     printf("The sum of odd arguments is %d\n", osum);

  return;
}

// convert string to number //
int st_to_int(char *str)
{
  int num =0;
  int i =0;
  while (str[i]!='\0') //c-string end with end line char//
  {
      num = 10 * num + (str[i] - '0');
      i++;
  }
  return num;
}
```

4.

```
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
#include <stdio.h>
#include <sys/stat.h>

int main()
{
     int in, out, i;  //file descriptors of files
     char c; //currently read character
     off_t offset; //current offset
     int size; //file size

     in = open("foo", O_RDONLY); //open input file
     umask(0); //clear mask
     out = open("palindrome", O_WRONLY|O_CREAT, 00600);
     //rw-------

     while (read(in, &c, 1) ==1)
          write(out, &c, 1);
     //set offset to end of input file & get file size
     size = lseek(in, -1, SEEK_END)+1;
     for (i =1; i<=size; i++)
     {
               read(in, &c, 1);//read each char of input file
               write(out, &c, 1);  //and write to output file
               lseek(in, -2, SEEK_CUR); //offset to previous char
     }
     //close open files
     close(in);
     close(out);
     exit(0);
}
```