**File Lock**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<fcntl.h>

#include<errno.h>

int main(int argc,char \*argv[]){

int fd;

char buffer[255];

struct flock fvar;

if(argc==1){

printf("usage %s filename \n",argv[0]);

return -1;

}

if(fd=open(argv[1],O\_RDWR)==-1){

perror("open");

exit(1);

}

fvar.l\_type=F\_WRLCK;

fvar.l\_whence=SEEK\_END;

fvar.l\_start=SEEK\_END-100;

fvar.l\_len=100;

printf("press enter to set lock\n");

getchar();

printf("trying to get lock\n");

if(fcntl(fd,F\_SETLK,&fvar)==-1){

fcntl(fd,F\_GETLK,&fvar);

printf("File is already locked by process pid: %d\n",fvar.l\_pid);

return -1;

}

printf("locked\n");

if(lseek(fd,SEEK\_END-50,SEEK\_END)==-1){

perror("lseek");

exit(1);

}

if(read(fd,buffer,100)==-1){

perror("read");

exit(1);

}

printf("data read from file\n");

puts(buffer);

printf("press enter to unlock");

getchar();

fvar.l\_type=F\_UNLCK;

fvar.l\_whence=SEEK\_SET;

fvar.l\_start=0;

fvar.l\_len=0;

if(fcntl(fd,F\_UNLCK,&fvar)==-1){

perror("fcntl");

exit(0);

}

printf("unlocked\n");

close(fd);

return 0;

}

/\*

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <errno.h>

int main(int argc,char \*argv[])

{

int fd;

char buffer[255];

struct flock fvar;

if(argc==1)

{

printf("usage: %s filename\n",argv[0]);

return -1;

}

if((fd=open(argv[1],O\_RDWR))==-1)

{

perror("open");

exit(1);

}

fvar.l\_type=F\_WRLCK;

fvar.l\_whence=SEEK\_END;

fvar.l\_start=SEEK\_END-100;

fvar.l\_len=100;

printf("press enter to set lock\n");

getchar();

printf("trying to get lock..\n");

if((fcntl(fd,F\_SETLK,&fvar))==-1)

{

fcntl(fd,F\_GETLK,&fvar);

printf("\nFile already locked by process (pid):\t%d\n",fvar.l\_pid);

return -1;

}

printf("locked\n");

if((lseek(fd,SEEK\_END-50,SEEK\_END))==-1)

{

perror("lseek");

exit(1);

}

if((read(fd,buffer,100))==-1)

{

perror("read");

exit(1);

}

printf("data read from file..\n");

puts(buffer);

printf("press enter to release lock\n");

getchar();

fvar.l\_type = F\_UNLCK;

fvar.l\_whence = SEEK\_SET;

fvar.l\_start = 0;

fvar.l\_len = 0;

if((fcntl(fd,F\_UNLCK,&fvar))==-1)

{

perror("fcntl");

exit(0);

}

printf("Unlocked\n");

close(fd);

return 0;

}\*/

**mv & rm**

int main(int argc, char\* argv[]) {

int input\_fd, output\_fd;

descriptors \*/

/\* Input and output file

/\* Are src and dest file name arguments missing \*/

if(argc != 3){

printf ("Usage: mv file1 file2");

return 1;

}

/\* Create input file descriptor \*/

input\_fd = link(argv [1], argv[2]);

if (input\_fd ==-1) {

perror ("link error");

return 2;

}

/\* Create output file descriptor \*/

output\_fd = unlink(argv[1]);

if(output\_fd ==-1){

perror("unlink");

return 3;

}

rmcommand

int main(int argc, char\* argv[]) {

output\_fd = unlink(argv[1]);

if(output\_fd ==-1){

perror("unlink error");

return 3;

}

}