#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <errno.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#define LINELEN 255

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

{

int fd = -1 ; //文件描述符

int rdsize = 0;

char buf[LINELEN+1]; //定義一個臨時緩衝區

if(1 >= argc){

printf("file name empty\n");

return 1;

}

fd = open(argv[1],O\_RDWR|O\_CREAT|O\_APPEND,0644); //可讀的寫的，wr-w--w--

if( fd < 0){ //return -1 代表開啟失敗 // 06444代表user可以讀可以寫

printf("open error [%d]\n",errno);

return 2;

}

buf[LINELEN] = '\0';

memset(buf,'\0',LINELEN); //'\0' == 0 把256個空間填入'\0' 清除用

while((rdsize = read(fd,buf,LINELEN)) > 0){ //每次讀（缓存區大小-1）的字節，直至讀的值為0

printf("%s",buf);

memset(buf,'\0',LINELEN); //每次都得進行清除

}

//printf("\n");

close(fd); //關閉描述符

return 0;

}

/\*

\*O\_RDWR代表在程式中可讀寫

\*O\_WRONLY 只寫打開

\*O\_CREATE則代表建立這個檔案

\*O\_RDONLY 只讀打開

\*以下可選項可以同時指定0個或多個，和必選項按位或起來作為flags參數。

\*O\_APPEND表示追加。如果文件已有內容，這次打開文件所寫的數據附加到文件的末尾而不覆蓋原來的內容。

\*O\_CREAT若此文件不存在則創建它。使用此選項時需要提供第三個參數mode，表示該文件的訪問權限。

\*O\_EXCL如果同時指定了O\_CREAT，並且文件已存在，則出錯返回。

\*O\_TRUNC 如果文件已存在，並且以只寫或可讀可寫方式打開，則將其長度截斷（Truncate）為0字節。

\*O\_NONBLOCK對於設備文件，以O\_NONBLOCK方式打開可以做非阻塞I/O（Nonblock I/O）。

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

fd = open(argv[1],O\_RDWR|O\_CREATE,0666); //可讀的寫，wr-w--w-- //0666代表任何人都可以可讀可寫

if(0 > fd){

printf("open error [%d]\n",errno);

return 2;

}

#include <unistd.h>

read()

#include <errno.h>

errno

#include <string.h>

memset()

\*/