/\*

\* pipe\_ls.c

\*/

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#define BSIZE 1024

int main() {

int fd[2];

pid\_t pid;

char buf[BSIZE];

if (pipe(fd) == -1) {

perror("pipe");

exit(1);

}

switch(pid=fork()) {

case -1:

perror("fork");

exit(1);

break;

/\* child \*/

case 0:

/\* Close read side, won't use it \*/

close(fd[0]); /\*//fd[0]=read child process , fd[1]=write child process

//0 stdin 標準輸入 ,1 stdout 標準輸出 ,2 stderr 標準錯誤\*/

dup2(fd[1],1);

if(execlp("ls","ls","-l",NULL)<0){

perror("execlp");

exit(1);

}

close(fd[1]);

exit(0);

break;

/\* parent \*/

default:

break;

}

/\* Close write side, won't use it \*/

close(fd[1]);

/\* Assumes a single string of \*/

/\* size less than BSIZE was written. \*/

memset(buf,'\0',BSIZE);

read( fd[0], buf, BSIZE);

printf("%s",buf); //puts(buf) ; puts 會把結束字元抓進來

close(fd[0]);

if (waitpid(-1, (int \*)NULL, 0) == -1) {

perror("waitpid");

exit(1);

}

return 0;

}