

Lezione 4 esercizi con soluzioni

Avendo come argomenti dei “binari”, si eseguono con `exec` ciascuno in un sottoprocesso (generando un figlio)

```
#include <stdlib.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

int main(int argc, char ** argv){
    for(int i = 1; i<argc;i++){
        int isChild = !fork(); //Generate child
        if(isChild){ //Launch executable only inside children
            char * argList[] = {argv[i],NULL}; //Define arguments
            execv(argv[i], argList); //Launch binary
        }
    }
    waitp(0,NULL,0); //Wait for all children
}
```

Avendo come argomenti dei “binari”, si eseguono con `exec` ciascuno in un sottoprocesso (generando un figlio), in più salvando i flussi di `stdout` e `stderr` in un unico file.

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

int main(int argc, char ** argv){
    //Open/create output file
    int outfile = open("/tmp/out.txt",O_RDWR | O_CREAT, S_IRUSR | S_IWUSR);

    //Open/create error file
    int errfile = open("/tmp/err.txt",
        O_RDWR | O_CREAT, S_IRUSR | S_IWUSR
    );

    dup2(outfile,1); //Redirect stdout to outputfile
    dup2(errfile,2); //Redirect stderr to errfile
    for(int i = 1; i<argc;i++){
        int isChild = !fork(); //Generate child
        if(isChild){ //Launch executable only inside children
            char * argList[] = {argv[i],NULL}; //Define arguments
            execv(argv[i], argList); //Launch binary
        }
    }
}
```

```

    }
}
waitpid(0,NULL,0); //Wait for all children
}

```

Dati due eseguibili come argomenti del tipo `ls` e `wc` si eseguono in due processi distinti: il primo deve generare uno *stdout* redirezionato su un file temporaneo, mentre il secondo deve essere lanciato solo quando il primo ha finito leggendo lo stesso file come *stdin*.

Ad esempio `./main ls wc` deve avere lo stesso effetto di `ls | wc`.

```

#include <stdio.h>
#include <fcntl.h>
#include <sys/wait.h>
#include <unistd.h>

void main() {
    char *args[2];
    int tmpFile = open("/tmp/temp.txt", O_RDWR | O_CREAT, S_IRUSR | S_IWUSR);
    int f=fork();
    if (f==0) {
        args[0]="/usr/bin/ls"; args[1]=NULL;
        dup2(tmpFile, 1);
        execvp(args[0],args);
    } else {
        wait(NULL);
        args[0]="/usr/bin/wc"; args[1]=NULL;
        dup2(tmpFile, 0); lseek(tmpFile, 0, SEEK_SET);
        execvp(args[0],args);
    };
    close(tmpFile);
}

```