Operating Systems(24802)
Project #1

Division of Computer Science
2017011912
Nam jung hoon

Contents

<Simple Shell Algorithm>

1. main loop 가 should_run 을 체크하며 반복함.

2. 한 줄의 입력을 받음.

3. 입력은 " "(공백)을 기준으로 각각 args 배열에 저장됨.
    3-1. 입력이 exit 라면 should_run 을 0 으로 바꿈.
    3-2. 입력이 '&' 로 끝난다면 back_is_ampersand 를 1 로 바꿈.

4. 자식프로세스 생성
    4.1. pid<0(생성 오류) 일 경우 Fork Failed 를 출력하며 프로그램 종료.
    4.2 pid==0(자식프로세서)일 경우 명령어 실행 (execvp 사용)
    4.3 pid>0 (부모프로세서)는 back_is_ampersand 변수를 기준으로
        자식 프로세스를 wait 하거나 wait 하지 않음.

5. 자식프로세서가 잘 생성되었다면 args 에 저장된 명령어들을 확인함.
    5.1 명령어에 '>' 이 포함된 경우 :
    '>' 이전까지의 입력에 대한 실행 결과를 '>' 이후 파일에 저장.(dup2 사용)
    5.2 명령어에 '<' 이 포함된 경우 :
    '<' 이후의 파일의 내용을 불러와 출력 (dup2 사용)
    5-3 명령어에 '|' 이 포함된 경우 :
    '|' 왼쪽의 실행 결과를 '|' 오른쪽의 프로세스로 넘김.(pipe 사용)

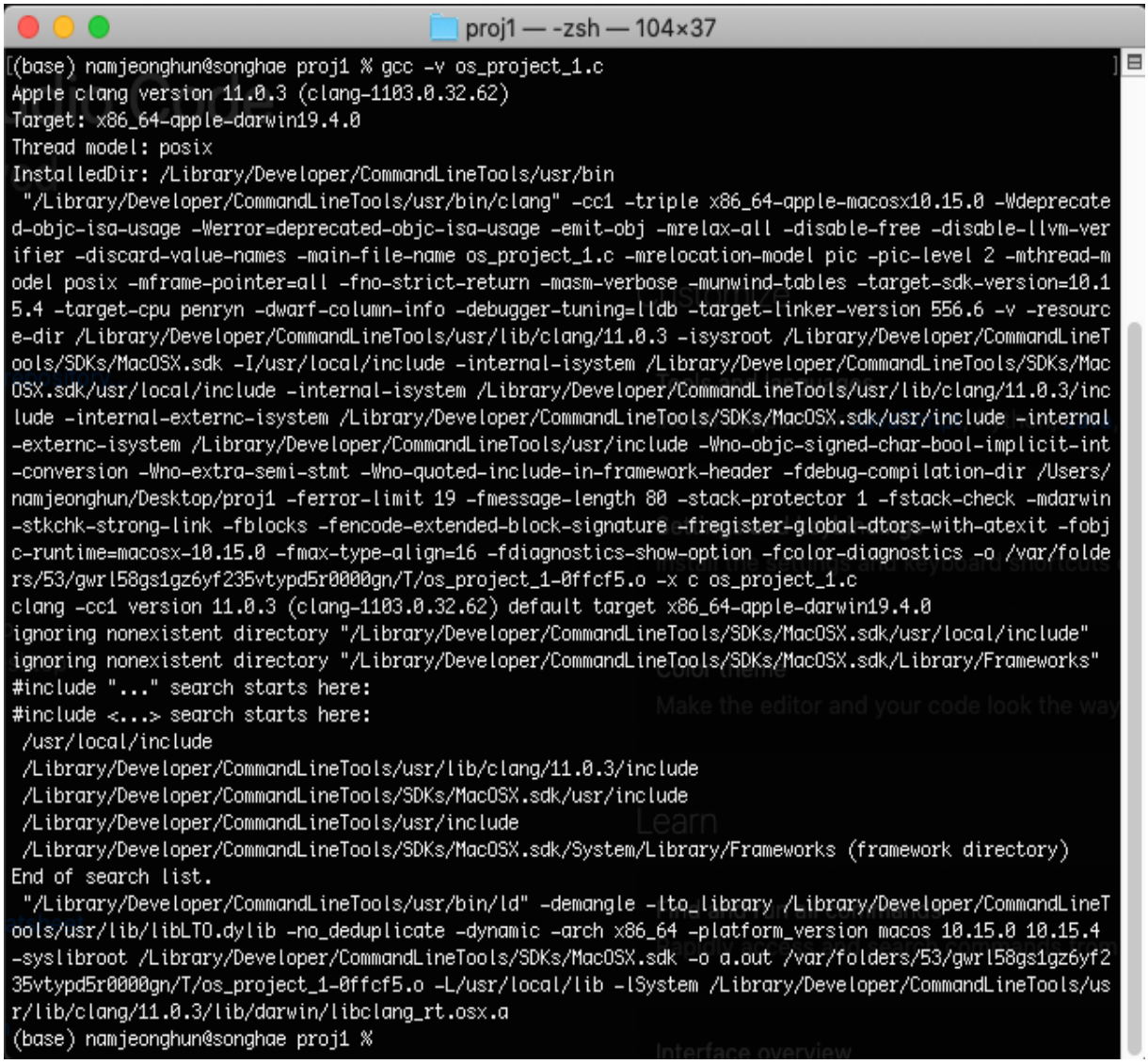
<Program source file>

```c
#include<stdio.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<unistd.h>
#include<string.h>
#include<stdlib.h>
#include<fcntl.h>
#define MAX_LINE 80  /* The maximum length command */

/* function to initialize shell */
void InitializeShell(){
    for(int i=5;i>0;i--){
        system("clear");
        printf("\n\n\n");
        printf("\t\t*******************************\n");
        printf("\t\t*                             *\n");
        printf("\t\t*                             *\n");
        printf("\t\t*      OS project #1          *\n");
        printf("\t\t*division of computer science*\n");
        printf("\t\t*        2017011912           *\n");
        printf("\t\t*       Namjeonghoon          *\n");
        printf("\t\t*                             *\n");
        printf("\t\t*    It starts in %d seconds  *\n",i);
        printf("\t\t*                             *\n");
        printf("\t\t*******************************\n");
        printf("\n\n\n");
        sleep(1);
        system("clear");
    }
}
/*
    function to check the command is "exit".
    if the command is "exit", return 1, or return 0.
    this return value will be used to determine whether to continue running the
main loop.
*/
int CheckExit(char** args){
    if(!strcmp(args[0],"exit")) return 1;
    else return 0;
}
/*
    function to take input.
    first,initialize args with (null).
    then recive a single line of input on my_buf,
    cut my_buf by ' ' or '\n'. and store them in the my_args.(args)
    the return value idicates the length of args.
*/
int TakeInput(char** my_args){
```

```
    char my_buf[MAX_LINE];
    int my_cnt = 0;
    char* my_ptr;
    fgets(my_buf,sizeof(my_buf),stdin);
    my_ptr = strtok(my_buf," \n");
    for(;my_ptr!=NULL;my_cnt++){
        my_args[my_cnt]=strdup(my_ptr);
        if(my_args[my_cnt][strlen(my_args[my_cnt])-1]=='\n'){
               my_args[my_cnt][strlen(my_args[my_cnt])-1]='\0';
            }
        my_ptr=strtok(NULL," \n");
    }
    return my_cnt;
}
/*
    function to check ampersand.
    if the end of command is &,
    the child process run in the background
*/
int CheckBack(char ** my_args,int my_cnt){
    if((my_cnt-1!=0)&&!strcmp(my_args[my_cnt-1],"&")){
           my_args[my_cnt-1]=NULL;
           return 1;
    }
    else return 0;
}
int main(){
    char *args[MAX_LINE/2+1]; /* command line arguments */
    int should_run = 1; /* flag to determine when to exit program */

    InitializeShell();
    while(should_run){
        int cnt,back_is_ampersand;
        pid_t pid;
        int fd[2];
        int status;
        memset(args,'\0',sizeof(args)); /* initialize args to (null) */

        printf("\nosh>");
        fflush(stdout);

        cnt = TakeInput(args);
        should_run = !CheckExit(args);
        back_is_ampersand = CheckBack(args,cnt);
        pid = fork();

        if(pid<0){
            fprintf(stderr, "Fork Failed");
            return 1;
        }
```

```
      else if(pid == 0){
         for(int i=0;i<cnt && args[i]!=NULL;i++){
            if(!strcmp(args[i],">")){ /* redirects the output of a command to a
file */
               args[i]=NULL;
               fd[0]=open(args[i+1],O_WRONLY|O_CREAT,0666);
               dup2(fd[0],STDOUT_FILENO);
               close(fd[0]);
            }
            else if(!strcmp(args[i],"<")){ /* redirects the input of a command
from a file */
               args[i]=NULL;
               fd[0]=open(args[i+1],O_RDONLY,0666);
               dup2(fd[0],STDIN_FILENO);
               close(fd[0]);
            }
            else if(!strcmp(args[i],"|")){ /* allow the output of one command to
serve as input to another using a pipe */
               args[i]=NULL;
               pipe(fd);
               pid_t pid2 = fork();
               if(pid2 < 0) fprintf(stderr, "Fork Failed");
               else if(pid2 == 0){
                  dup2(fd[1],1);
                  close(fd[0]);
                  execvp(args[0],args);
               }
               else{
                  dup2(fd[0],0);
                  close(fd[1]);
                  execvp(args[i+1],&args[i+1]);
               }
            }
         }
         execvp(args[0],args);
      }
      else{
         if(back_is_ampersand){
            waitpid(pid,&status,WNOHANG);
            printf("Child Complete");
         }
         else{
            waitpid(pid,&status,0);
            printf("Child Complete");
         }
      }
   }
   return 0;
}
```

<Compilation Process>



```
(base) namjeonghun@songhae proj1 % gcc -v os_project_1.c
Apple clang version 11.0.3 (clang-1103.0.32.62)
Target: x86_64-apple-darwin19.4.0
Thread model: posix
InstalledDir: /Library/Developer/CommandLineTools/usr/bin
 "/Library/Developer/CommandLineTools/usr/bin/clang" -cc1 -triple x86_64-apple-
macosx10.15.0 -Wdeprecated-objc-isa-usage -Werror=deprecated-objc-isa-usage -emit-
obj -mrelax-all -disable-free -disable-llvm-verifier -discard-value-names -main-
file-name os_project_1.c -mrelocation-model pic -pic-level 2 -mthread-model posix -
mframe-pointer=all -fno-strict-return -masm-verbose -munwind-tables -target-sdk-
version=10.15.4 -target-cpu penryn -dwarf-column-info -debugger-tuning=lldb -
target-linker-version 556.6 -v -resource-dir
/Library/Developer/CommandLineTools/usr/lib/clang/11.0.3 -isysroot
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk -I/usr/local/include -internal-
isystem /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/local/include -
internal-isystem /Library/Developer/CommandLineTools/usr/lib/clang/11.0.3/include -
internal-externc-isystem
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include -internal-externc-
isystem /Library/Developer/CommandLineTools/usr/include -Wno-objc-signed-char-bool-
implicit-int-conversion -Wno-extra-semi-stmt -Wno-quoted-include-in-framework-
header -fdebug-compilation-dir /Users/namjeonghun/Desktop/proj1 -ferror-limit 19 -
fmessage-length 80 -stack-protector 1 -fstack-check -mdarwin-stkchk-strong-link -
fblocks -fencode-extended-block-signature -fregister-global-dtors-with-atexit -
fobjc-runtime=macosx-10.15.0 -fmax-type-align=16 -fdiagnostics-show-option -fcolor-
diagnostics -o /var/folders/53/gwrl58gs1gz6yf235vtypd5r0000gn/T/os_project_1-
0ffcf5.o -x c os_project_1.c
clang -cc1 version 11.0.3 (clang-1103.0.32.62) default target x86_64-apple-
darwin19.4.0
ignoring nonexistent directory
"/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/local/include"
ignoring nonexistent directory
"/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/Library/Frameworks"
#include "..." search starts here:
#include <...> search starts here:
 /usr/local/include
 /Library/Developer/CommandLineTools/usr/lib/clang/11.0.3/include
 /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include
 /Library/Developer/CommandLineTools/usr/include
 /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/System/Library/Frameworks
(framework directory)
End of search list.
 "/Library/Developer/CommandLineTools/usr/bin/ld" -demangle -lto_library
/Library/Developer/CommandLineTools/usr/lib/libLTO.dylib -no_deduplicate -dynamic -
arch x86_64 -platform_version macos 10.15.0 10.15.4 -syslibroot
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk -o a.out
/var/folders/53/gwrl58gs1gz6yf235vtypd5r0000gn/T/os_project_1-0ffcf5.o -
L/usr/local/lib -lSystem
/Library/Developer/CommandLineTools/usr/lib/clang/11.0.3/lib/darwin/libclang_rt.osx
.a
```

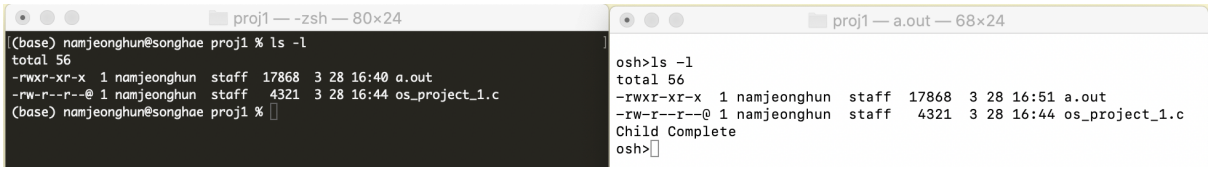
<Deliverables and Descriptions>

zshell(mac os) my_shell

1. 명령어 + 옵션

ls -l

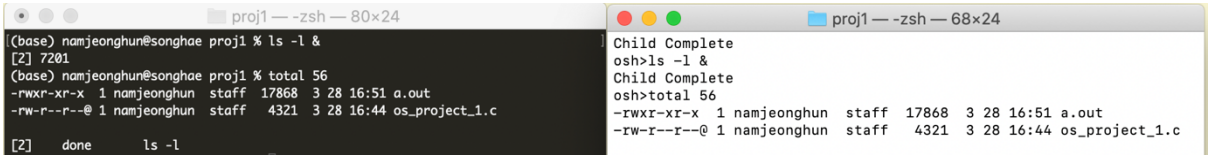
실행결과 :



2. 명령어 + 옵션 &

ls -l &

실행결과 :



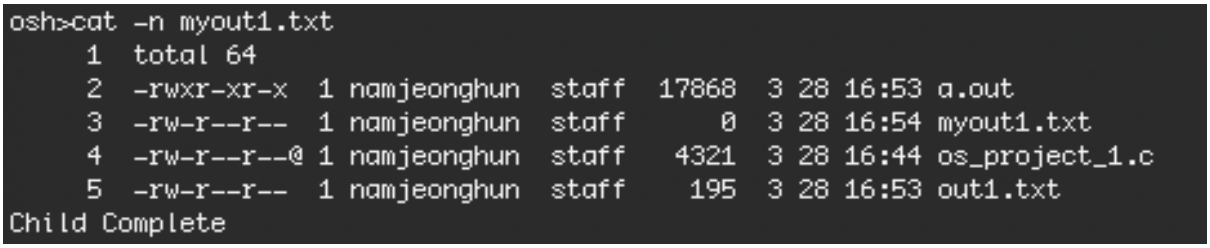
3. 명령어 + 옵션 > 파일명

ls -l > myout1.txt

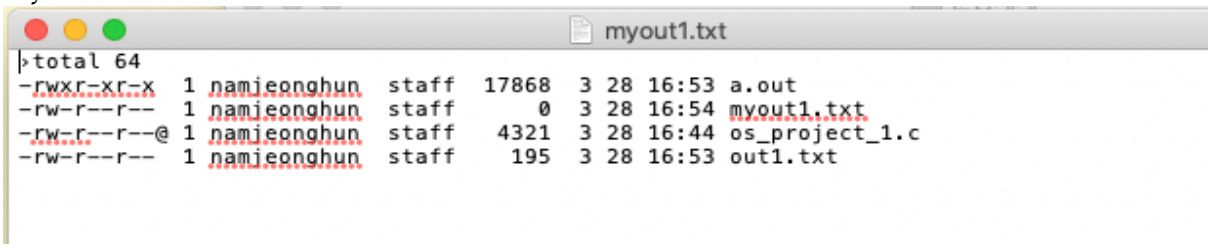
실행결과 :

cat -n myout1.txt

실행결과 :



myout1.txt 파일 :



4. 명령어 + 옵션 > 파일명 &

ls -l > myout2.txt &

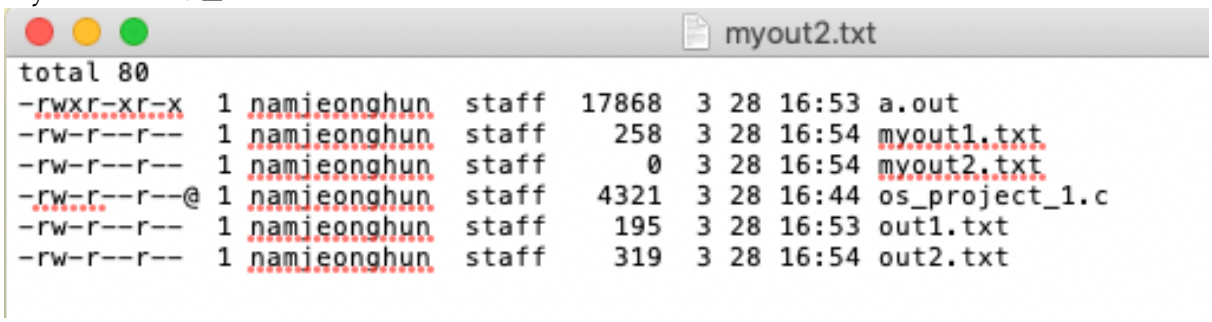
실행결과 :

cat – n myout2.txt

실행결과 :

```
osh>cat -n myout2.txt
     1  total 80
     2  -rwxr-xr-x  1 namjeonghun  staff  17868  3 28 16:53 a.out
     3  -rw-r--r--  1 namjeonghun  staff    258  3 28 16:54 myout1.txt
     4  -rw-r--r--  1 namjeonghun  staff      0  3 28 16:54 myout2.txt
     5  -rw-r--r--@ 1 namjeonghun  staff   4321  3 28 16:44 os_project_1.c
     6  -rw-r--r--  1 namjeonghun  staff    195  3 28 16:53 out1.txt
     7  -rw-r--r--  1 namjeonghun  staff    319  3 28 16:54 out2.txt
Child Complete
```

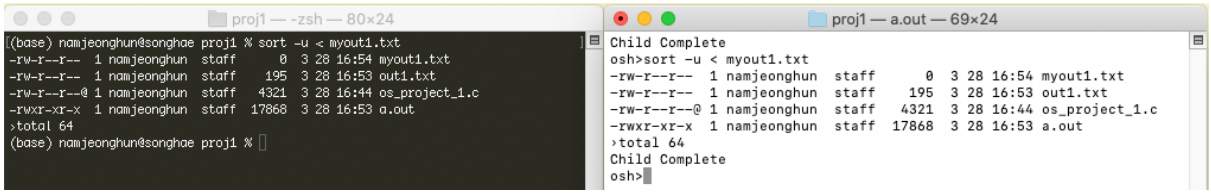
myout2.txt 파일 :



5. 명령어 + 옵션 < 파일명

sort – u < myout1.txt (3 의 파일과 동일)

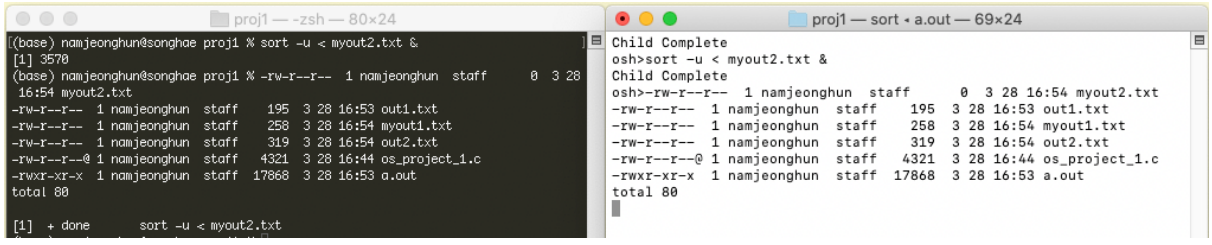
실행결과 :



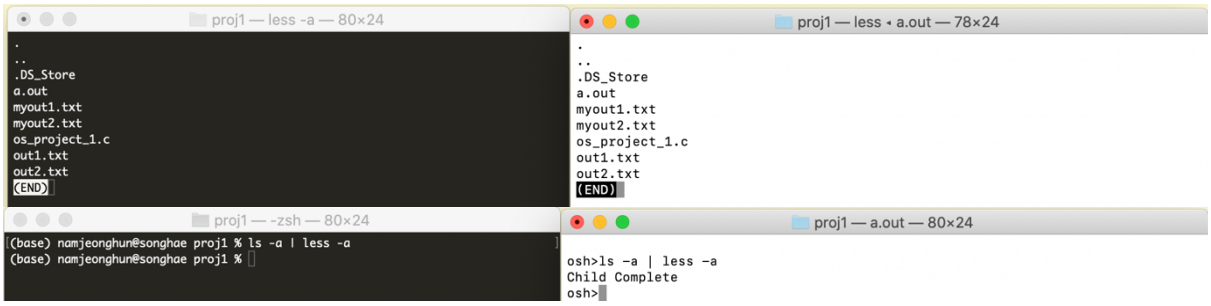
6. 명령어 + 옵션 < 파일명 &

sort -u < myout2.txt & (4 의 파일과 동일)

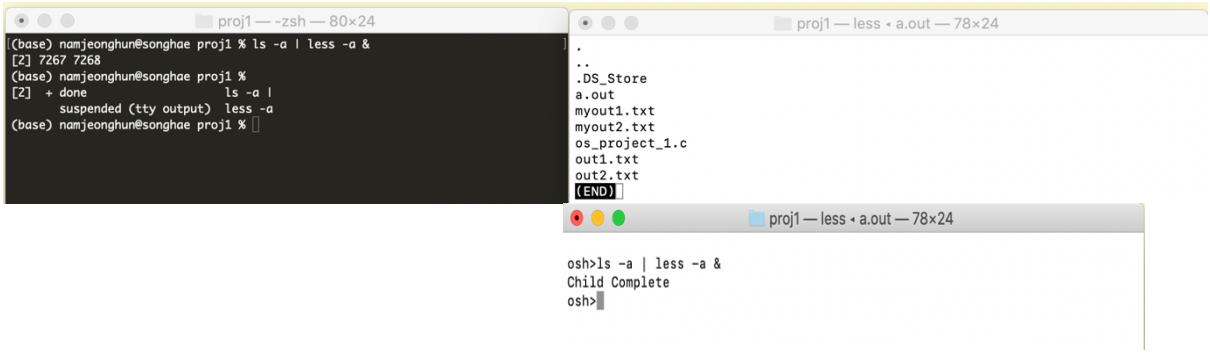
실행결과 :



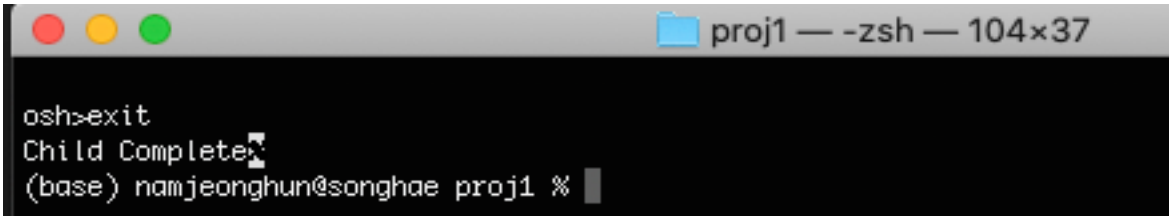
7. 명령어 + 옵션 | 명령어 + 옵션
ls – a | less – a
실행결과 :



8. 명령어 + 옵션 | 명령어 + 옵션 &
ls -a | less -a &
실행결과 :



9. 종료 커맨드
exit
실행결과 :



10. 종료 커맨드 &
Exit &
실행결과 :