Q1

#include <unistd.h>

#include <stdio.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <dirent.h>

#include <string.h>

int main()

{

DIR \*mydir;

char \*c;

int i;

struct dirent \*myfile;

struct stat fileStat;

mydir = opendir(".");

stat(".", &fileStat);

while ((myfile = readdir(mydir)) != NULL)

{

stat(myfile->d\_name, &fileStat);

if (strcmp(myfile->d\_name, ".") != 0 && strcmp(myfile->d\_name, "..") != 0)

{

printf("%s\n", myfile->d\_name);

printf("File Permission:\n");

printf((S\_ISDIR(fileStat.st\_mode)) ? "d" : "-");

printf((fileStat.st\_mode & S\_IRUSR) ? "r" : "-");

printf((fileStat.st\_mode & S\_IWUSR) ? "w" : "-");

printf((fileStat.st\_mode & S\_IXUSR) ? "x" : "-");

printf((fileStat.st\_mode & S\_IRGRP) ? "r" : "-");

printf((fileStat.st\_mode & S\_IWGRP) ? "w" : "-");

printf((fileStat.st\_mode & S\_IXGRP) ? "x" : "-");

printf((fileStat.st\_mode & S\_IROTH) ? "r" : "-");

printf((fileStat.st\_mode & S\_IWOTH) ? "w" : "-");

printf((fileStat.st\_mode & S\_IXOTH) ? "x" : "-");

printf("\n");

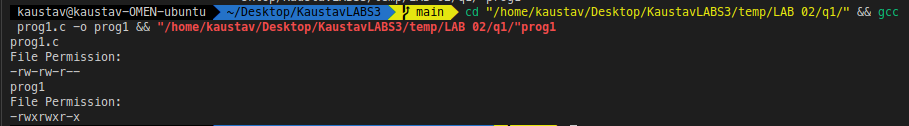
}

}

closedir(mydir);

return 0;

}



Q2

#include <unistd.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <dirent.h>

void printdir(char \*dir, int depth)

{

DIR \*dp;

struct dirent \*entry;

struct stat statbuf;

if ((dp = opendir(dir)) == NULL)

{

fprintf(stderr, "cannot open directory: %s\n", dir);

return;

}

chdir(dir);

while ((entry = readdir(dp)) != NULL)

{

lstat(entry->d\_name, &statbuf);

if (S\_ISDIR(statbuf.st\_mode))

{

if (strcmp(".", entry->d\_name) == 0 || strcmp("..", entry->d\_name) == 0)

continue;

printf("%\*s%s/\n", depth, " ", entry->d\_name);

printdir(entry->d\_name, depth + 4);

}

else

printf("%\*s%s/\n", depth, " ", entry->d\_name);

}

chdir("..");

closedir(dp);

}

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

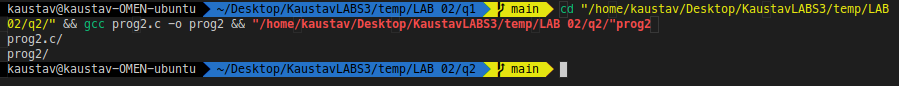
{

/\* code \*/

printdir(".", 0);

exit(0);

}



Q3

#include <unistd.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <dirent.h>

void printdir(char \*dir, int depth)

{

DIR \*dp;

struct dirent \*entry;

struct stat statbuf;

if ((dp = opendir(dir)) == NULL)

{

fprintf(stderr, "cannot open directory: %s\n", dir);

return;

}

chdir(dir);

while ((entry = readdir(dp)) != NULL)

{

lstat(entry->d\_name, &statbuf);

if (S\_ISDIR(statbuf.st\_mode))

{

if (strcmp(".", entry->d\_name) == 0 || strcmp("..", entry->d\_name) == 0)

continue;

printf("%\*s%s/\n", depth, " ", entry->d\_name);

printdir(entry->d\_name, depth + 4);

}

else

printf("%\*s%s/\n", depth, " ", entry->d\_name);

}

chdir("..");

closedir(dp);

}

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

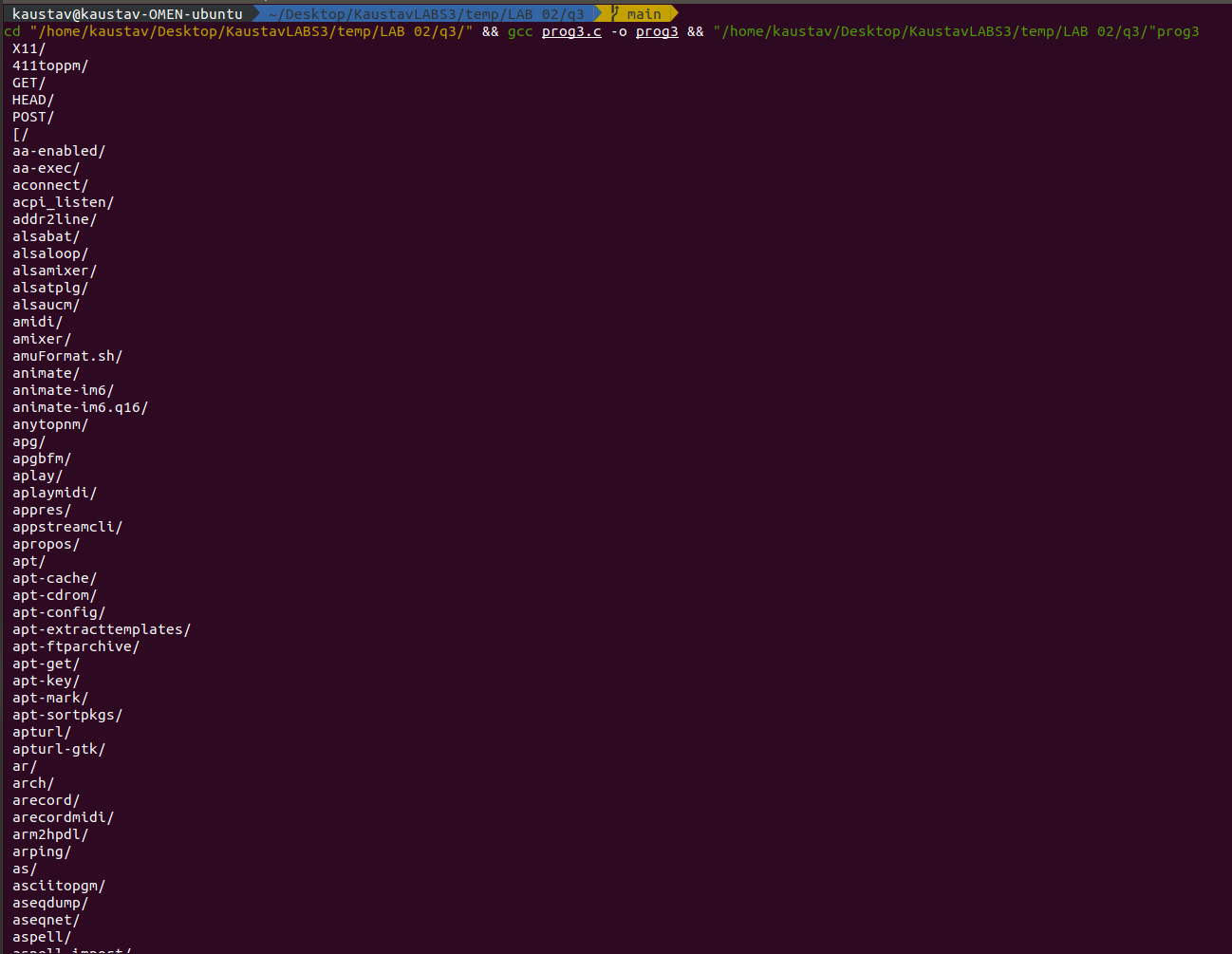
{

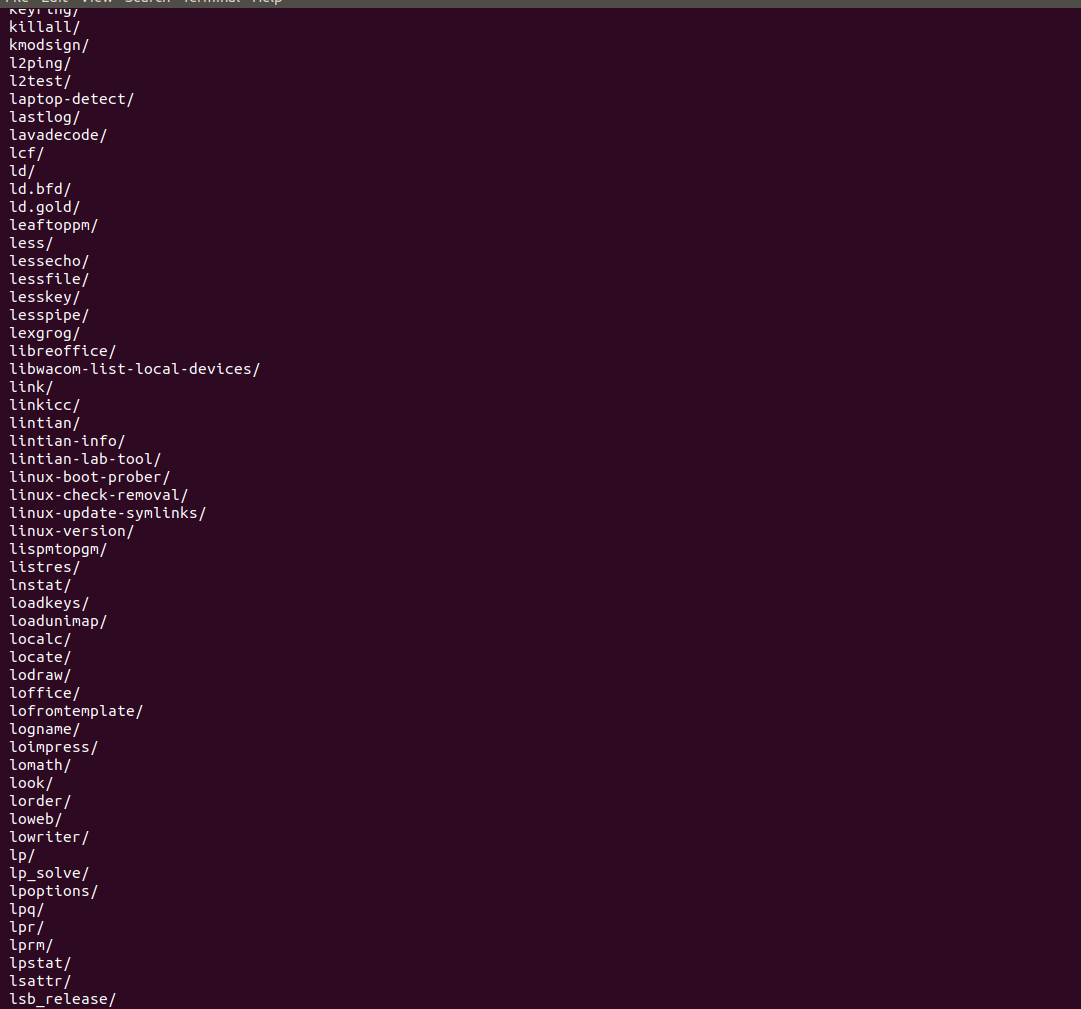
/\* code \*/

printdir("/usr/bin", 0);

exit(0);

}





Q4

#include <unistd.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <dirent.h>

void printdir(char \*dir, int depth)

{

DIR \*dp;

struct dirent \*entry;

struct stat statbuf;

if ((dp = opendir(dir)) == NULL)

{

fprintf(stderr, "cannot open directory: %s\n", dir);

return;

}

chdir(dir);

while ((entry = readdir(dp)) != NULL)

{

lstat(entry->d\_name, &statbuf);

if (S\_ISDIR(statbuf.st\_mode))

{

if (strcmp(".", entry->d\_name) == 0 || strcmp("..", entry->d\_name) == 0)

continue;

printf("%\*s%s/\n", depth, " ", entry->d\_name);

printdir(entry->d\_name, depth + 4);

}

else

printf("%\*s%s/\n", depth, " ", entry->d\_name);

}

chdir("..");

closedir(dp);

}

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

{

/\* code \*/

printdir("/dev", 0);

exit(0);

}

