#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

int main() {

char fileName[] = "test\_file.txt";

int fileDescriptor = open(fileName, O\_CREAT | O\_RDWR, 0644);

if (fileDescriptor == -1) {

perror("open");

exit(EXIT\_FAILURE);

}

char content[] = "Hello, world!";

ssize\_t bytesWritten = write(fileDescriptor, content, sizeof(content) - 1);

if (bytesWritten == -1) {

perror("write");

exit(EXIT\_FAILURE);

}

off\_t offset = lseek(fileDescriptor, 7, SEEK\_SET);

if (offset == -1) {

perror("lseek");

exit(EXIT\_FAILURE);

}

char buffer[10];

ssize\_t bytesRead = read(fileDescriptor, buffer, sizeof(buffer) - 1);

if (bytesRead == -1) {

perror("read");

exit(EXIT\_FAILURE);

}

buffer[bytesRead] = '\0';

printf("从偏移量 %ld 读取了 %ld 字节: %s\n", offset, bytesRead, buffer);

if (close(fileDescriptor) == -1) {

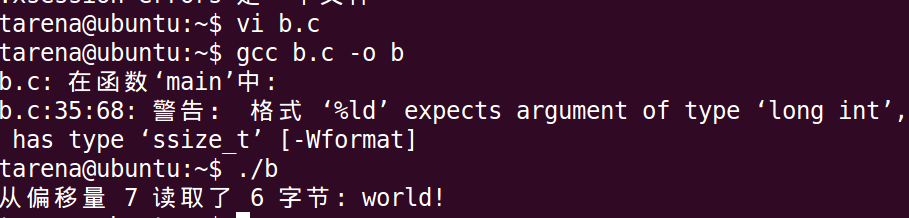
perror("close");

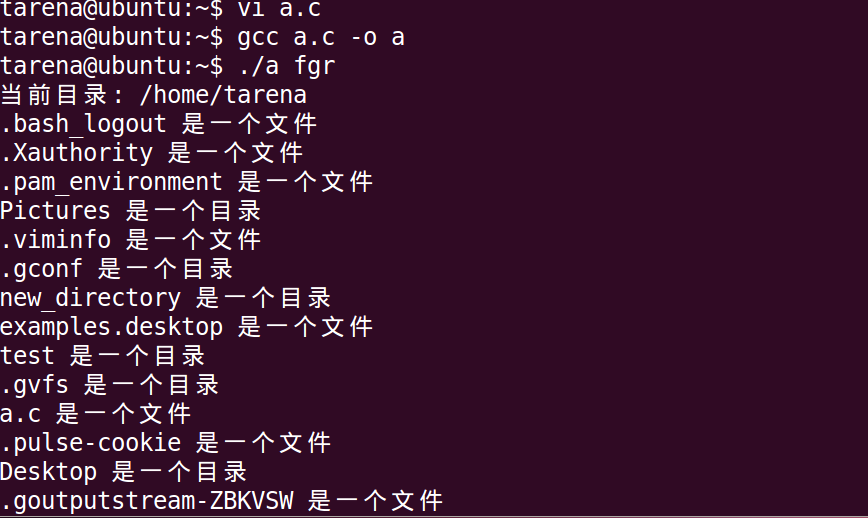
exit(EXIT\_FAILURE);

}

return 0;

}





#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <dirent.h>

int main() {

*// 创建新目录*

char dirName[] = "new\_directory";

int result = mkdir(dirName, 0777);

if (result == -1) {

perror("mkdir");

exit(EXIT\_FAILURE);

}

*// 获取当前目录并输出打印*

char currentDir[1024];

if (getcwd(currentDir, sizeof(currentDir)) != NULL) {

printf("当前目录: %s\n", currentDir);

} else {

perror("getcwd");

exit(EXIT\_FAILURE);

}

*// 打开目录并读取目录信息*

DIR \*dir = opendir(".");

if (dir == NULL) {

perror("opendir");

exit(EXIT\_FAILURE);

}

*// 利用stat函数读取当前目录的状态信息，并输出判断结果*

struct dirent \*entry;

while ((entry = readdir(dir)) != NULL) {

struct stat fileInfo;

if (stat(entry->d\_name, &fileInfo) == -1) {

perror("stat");

exit(EXIT\_FAILURE);

}

if (S\_ISDIR(fileInfo.st\_mode)) {

printf("%s 是一个目录\n", entry->d\_name);

} else {

printf("%s 是一个文件\n", entry->d\_name);

}

}

*// 关闭目录*

closedir(dir);

return 0;

}