

OS-SYSTEM CALLS DIRECTORY RELATED

2019103562

- ❖ OPENDIR()
- ❖ CLOSEDIR()
- ❖ LSTAT() AND LSTAT64()
- ❖ STAT AND STAT64()
- ❖ DIRCNTL()
- ❖ FTW()
- ❖ READDIR_R() AND READDIR64_R()
- ❖ REWINDDIR()
- ❖ SEEKDIR()
- ❖ TELLDIR()
- ❖ ERRNO

```
[s2019103562@centos8-linux Mon Mar 15 09:55 AM ~]$ vim opendir.c
[s2019103562@centos8-linux Mon Mar 15 09:56 AM ~]$ gcc opendir.c -o opendir
[s2019103562@centos8-linux Mon Mar 15 09:56 AM ~]$ ./opendir
File 1:.
File 2:..
File 3:lab1pract.c
File 4:1a.txt
File 5:1c.txt
File 6:sample.c
File 7:sample2.c
File 8:eg.txt
File 9:samp.txt
File 10:test.txt
File 11:emp.txt
File 12:empt.txt
File 13:file.txt
File 14:books-authors.txt
File 15:test.sh
File 16:New.c
File 17:NewFile.c
[s2019103562@centos8-linux Mon Mar 15 09:56 AM ~]$ cat opendir.c
```

```
[s2019103562@centos8-linux Mon Mar 15 09:56 AM ~]$ cat opendir.c
#include<stdio.h>
#include<stdlib.h>
#include<dirent.h>
void main(){
    struct dirent* direntp;
    DIR* dp;
    int i=1;
    dp=opendir("lab01");
    if(dp==NULL)
        printf("Can't open the directory\n");
    else{
        for(;;){
            direntp=readdir(dp);
            if(direntp==NULL)
                break;
            printf("File %d:%s\n",i++,direntp->d_name);
        }
    }
    closedir(dp);
}
[s2019103562@centos8-linux Mon Mar 15 09:56 AM ~]$
```

```
[s2019103562@centos8-linux Mon Mar 15 10:48 AM ~]$ vim open.c
[s2019103562@centos8-linux Mon Mar 15 10:51 AM ~]$ gcc open.c -o open
[s2019103562@centos8-linux Mon Mar 15 10:51 AM ~]$ ./open
Listing the files of subdirectory
File:1 is .
File:2 is ..
File:3 is hello.txt
File:4 is Sample.c
[s2019103562@centos8-linux Mon Mar 15 10:51 AM ~]$ cat open.c
#include<stdio.h>
#include<stdlib.h>
#include<dirent.h>
#include<unistd.h>
void main(){
    DIR* dp;
    int i=1;
    struct dirent* ptr;
    dp=opendir("lab01/inner");
    if(dp==NULL)
        printf("can't open the directory\n");
    else{
        printf("Listing the files of subdirectory\n");
        while((ptr=readdir(dp))!=NULL){
            printf("File:%d is %s\n",i++,ptr->d_name);
        }
    }
    closedir(dp);
}
[s2019103562@centos8-linux Mon Mar 15 10:51 AM ~]$
```

2. READDIR_R() AND READDIR64_R():

```
[s2019103562@centos8-linux Mon Mar 15 09:14 AM ~]$ ./readdir_r lab02
Directory entry is 1:..
Directory entry is 2:..
Directory entry is 3:arr.sh
Directory entry is 4:decmaking.sh
Directory entry is 5:forloop.sh
Directory entry is 6:pat.txt
Directory entry is 7:sample.txt
Directory entry is 8:switch.sh
Directory entry is 9:until.sh
Directory entry is 10:while.sh
Directory entry is 11:stdnt.sh
[s2019103562@centos8-linux Mon Mar 15 09:14 AM ~]$ cat readdir_r.c
#include<stdio.h>
#include<stdlib.h>
#include<dirent.h>
#include<sys/types.h>
int main(int argc,char* argv[]){
    DIR* d;
    struct dirent de, *dep;
    char* mydir;
    int i;
    if(argc==1)
        mydir=".";
    else
        mydir=argv[1];
    d=opendir(mydir);
    if(d==NULL)
        printf("Can't open the directory\n");
    i=1;
    while(readdir_r(d,&de,&dep)==0 && dep!=NULL){
        printf("Directory entry is %d:%s\n",i++,de.d_name);
    }
    closedir(d);
    return 0;
}
[s2019103562@centos8-linux Mon Mar 15 09:15 AM ~]$
```

```
[s2019103562@centos8-linux Mon Mar 15 09:15 AM ~]$ ./readdir_r
Directory entry is 1:..
Directory entry is 2:..
Directory entry is 3:.mozilla
Directory entry is 4:.bash_logout
Directory entry is 5:.bash_profile
Directory entry is 6:.bashrc
Directory entry is 7:.config
Directory entry is 8:.esd_auth
Directory entry is 9:.bash_history
Directory entry is 10:lab01
Directory entry is 11:lab02
Directory entry is 12:lab05
Directory entry is 13:.viminfo
Directory entry is 14:oradiag_s2019103562
Directory entry is 15:lab04
Directory entry is 16:opendir.c
Directory entry is 17:.fork.c.swo
Directory entry is 18:calab01
Directory entry is 19:opendir
Directory entry is 20:readdir_r.c
Directory entry is 21:lab03
Directory entry is 22:mkdir
Directory entry is 23:.java
Directory entry is 24:.Xauthority
Directory entry is 25:readdir_r
[s2019103562@centos8-linux Mon Mar 15 09:15 AM ~]$
```

Readdir64_r() is for large file support.

3.LSTAT() AND LSTAT64():

```
[s2019103562@centos8-linux Mon Mar 15 12:15 PM ~]$ vim lstat.c
[s2019103562@centos8-linux Mon Mar 15 12:17 PM ~]$ gcc lstat.c -o lstat
[s2019103562@centos8-linux Mon Mar 15 12:17 PM ~]$ ./lstat
[s2019103562@centos8-linux Mon Mar 15 12:17 PM ~]$ gcc lstat.c -o lstat
[s2019103562@centos8-linux Mon Mar 15 12:17 PM ~]$ ./lstat rewinddir.c
rewinddir.c is not a symbolic link
The size of the file:629
rw-r--r--
Number of links:1
File inode:537326781
[s2019103562@centos8-linux Mon Mar 15 12:17 PM ~]$ cat lstat.c

#include<stdio.h>
#include<stdlib.h>
#include<sys/stat.h>
#include<sys/types.h>
#include<unistd.h>
int main(int argc,char* argv[]){
    int n;
    struct stat buf;
    for(n=1;n<argc;n++){
        if(lstat(argv[n],&buf)==-1){
            printf("Error in accessing the file\n");
            return 0;
        }
        if(S_ISLNK(buf.st_mode))
            printf("%s is a symbolic link\n",argv[n]);
        else
            printf("%s is not a symbolic link\n",argv[n]);
        printf("The size of the file:%d\n",buf.st_size);
        printf((S_IRUSR & buf.st_mode) ? "r" : "-");
        printf((S_IWUSR & buf.st_mode) ? "w" : "-");
        printf((S_IXUSR & buf.st_mode) ? "x" : "-");
        printf((S_IRGRP & buf.st_mode) ? "r" : "-");
        printf((S_IWGRP & buf.st_mode) ? "w" : "-");
        printf((S_IXGRP & buf.st_mode) ? "x" : "-");
        printf((S_IROTH & buf.st_mode) ? "r" : "-");
        printf((S_IWOTH & buf.st_mode) ? "w" : "-");
        printf((S_IXOTH & buf.st_mode) ? "x" : "-");
        printf("\n");
        printf("Number of links:%d\n",buf.st_nlink);
        printf("File inode:%d\n",buf.st_ino);
    }
    return 0;
}
```

Lstat64() is for large file support.

4.REWINDDIR():

```
[s2019103562@centos8-linux Mon Mar 15 09:54 AM ~]$cat rewinddir.c
#include<stdio.h>
#include<stdlib.h>
#include<dirent.h>
#include<sys/types.h>
#include<sys/stat.h>
int main(){
    DIR* dp;
    struct dirent* direntp;
    int file;
    int i=1,j=1;
    dp=opendir("lab01");
    if(dp==NULL){
        printf("Error in opening the file\n");
        return 0;
    }
    else{
        printf("Old directory listing\n");
        while((direntp=readdir(dp))!=NULL){
            printf("File %d:%s\n",i++,direntp->d_name);
        }
        file=creat("lab01/NewFile.c",S_IRUSR | S_IWUSR | S_IXUSR);
        close(file);
        rewinddir(dp);
        printf("New directory listing\n");
        while((direntp=readdir(dp))!=NULL){
            printf("File %d:%s\n",j++,direntp->d_name);
        }
        closedir(dp);
    }
}
[s2019103562@centos8-linux Mon Mar 15 09:55 AM ~]$
```

```

[s2019103562@centos8-linux Mon Mar 15 09:54 AM ~]$ ./rewinddir
Old directory listing
File 1:..
File 2:...
File 3:lab1pract.c
File 4:1a.txt
File 5:1c.txt
File 6:sample.c
File 7:sample2.c
File 8:eg.txt
File 9:samp.txt
File 10:test.txt
File 11:emp.txt
File 12:empt.txt
File 13:file.txt
File 14:books-authors.txt
File 15:test.sh
File 16:New.c
New directory listing
File 1:..
File 2:...
File 3:lab1pract.c
File 4:1a.txt
File 5:1c.txt
File 6:sample.c
File 7:sample2.c
File 8:eg.txt
File 9:samp.txt
File 10:test.txt
File 11:emp.txt
File 12:empt.txt
File 13:file.txt
File 14:books-authors.txt
File 15:test.sh
File 16:New.c
File 17:NewFile.c
[s2019103562@centos8-linux Mon Mar 15 09:54 AM ~]$ █

```

5.SEEKDIR():

```

-----
[s2019103562@centos8-linux Mon Mar 15 09:51 AM ~]$ vim seekdir.c
[s2019103562@centos8-linux Mon Mar 15 09:52 AM ~]$ gcc seekdir.c -o seekdir
[s2019103562@centos8-linux Mon Mar 15 09:52 AM ~]$ ./seekdir
File name:., Offset:10
File name:.., Offset:12
File name:arr.sh, Offset:15
File name:decmaking.sh, Offset:18
File name:forloop.sh, Offset:21
File name:pat.txt, Offset:24
File name:sample.txt, Offset:27
File name:switch.sh, Offset:30
File name:until.sh, Offset:33
File name:while.sh, Offset:36
File name:stdnt.sh, Offset:512
Reading the dircectory again with the offset of 5,that is skipping the first 5 filenames
File name:pat.txt, Offset:24
File name:sample.txt, Offset:27
File name:switch.sh, Offset:30
File name:until.sh, Offset:33
File name:while.sh, Offset:36
File name:stdnt.sh, Offset:512
[s2019103562@centos8-linux Mon Mar 15 09:52 AM ~]$ cat seekdir.c
#include <stdio.h>
#

```

```
[s2019103562@centos8-linux Mon Mar 15 09:52 AM ~]$ cat seekdir.c
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<dirent.h>
#include<unistd.h>
void main(){
    struct dirent* ptr;
    int offset,offset_5,i=0;
    DIR* dir;
    dir=opendir("lab02");
    if(dir==NULL)
        printf("Can't open the directory\n");
    else{
        while((ptr=readdir(dir))!=NULL){
            offset=telldir(dir);
            if(++i==5)
                offset_5=offset;
            printf("File name:%s, Offset:%d\n",ptr->d_name,offset);
        }
        seekdir(dir,offset_5);
        printf("Reading the directory again with the offset of 5,that is skipping the first 5 filenames\n");
        while((ptr=readdir(dir))!=NULL){
            offset=telldir(dir);
            printf("File name:%s, Offset:%d\n",ptr->d_name,offset);
        }
    }
    closedir(dir);
}
[s2019103562@centos8-linux Mon Mar 15 09:52 AM ~]$
```

6.TELLDIR():

```
[s2019103562@centos8-linux Mon Mar 15 09:49 AM ~]$ vim telldir.c
[s2019103562@centos8-linux Mon Mar 15 09:50 AM ~]$ gcc telldir.c -o telldir
[s2019103562@centos8-linux Mon Mar 15 09:50 AM ~]$ ./telldir
File 10:.
File 12:...
File 15:logicop.sh
File 18:break.sh
File 21:continue.sh
File 24:array.sh
File 27:function.sh
File 30:vow.sh
File 33:readtxt.sh
File 36:writetxt.sh
File 39:file.sh
File 42:filen.txt
File 45:ftest.sh
File 512:1a.sh
[s2019103562@centos8-linux Mon Mar 15 09:50 AM ~]$ cat telldir.c
#include<stdio.h>
#include<stdlib.h>
#include<dirent.h>
#include<sys/types.h>
#include<unistd.h>
void main(){
    DIR* dp;
    struct dirent* ptr;
    int offset;
    dp=opendir("lab03");
    if(dp==NULL)
        printf("Can't open the directory\n");
    else{
        while((ptr=readdir(dp))!=NULL){
            offset=telldir(dp);
            printf("File %d:%s\n",offset,ptr->d_name);
        }
    }
    closedir(dp);
}
[s2019103562@centos8-linux Mon Mar 15 09:50 AM ~]$
```

7.STAT() AND STAT64():

```
[s2019103562@centos8-linux Mon Mar 15 10:11 AM ~]$ vim stat.c
[s2019103562@centos8-linux Mon Mar 15 10:12 AM ~]$ gcc stat.c -o stat
[s2019103562@centos8-linux Mon Mar 15 10:12 AM ~]$ ./stat
Enter the filename
rewinddir.c
File size:629
Readable by User
Writable by User
Not executable by user
Readable by Group
Not writable by group
Not executable by group
Not writable by others
Readable by Others
Not executable by others
[s2019103562@centos8-linux Mon Mar 15 10:13 AM ~]$ cat stat.c
```

```
[s2019103562@centos8-linux Mon Mar 15 10:13 AM ~]$ cat stat.c
#include<stdio.h>
#include<stdlib.h>
#include<sys/stat.h>
#include<sys/types.h>
#include<unistd.h>
int main(){
    struct stat buf;
    char file[30];
    printf("Enter the filename\n");
    scanf("%s",file);
    if(stat(file,&buf)==-1){
        printf("Error occured\n");
        return 0;
    }
    else{
        printf("File size:%d\n",buf.st_size);
        if(buf.st_mode & S_IRUSR)
            printf("Readable by User\n");
        else
            printf("Not readable by user\n");
        if(buf.st_mode & S_IWUSR)
            printf("Writable by User\n");
        else
            printf("Not writable by user\n");
        if(buf.st_mode & S_IXUSR)
            printf("Executable by User\n");
        else
            printf("Not executable by user\n");
        if(buf.st_mode & S_IRGRP)
            printf("Readable by Group\n");
        else
            printf("Not readable by group\n");
    }
}
```



```

        if(buf.st_mode & S_IWGRP)
            printf("Writable by Group\n");
        else
            printf("Not writable by group\n");
        if(buf.st_mode & S_IXGRP)
            printf("Executable by Group\n");
        else
            printf("Not executable by group\n");
        if(buf.st_mode & S_IWOTH)
            printf("Writable by Others\n");
        else
            printf("Not writable by others\n");
        if(buf.st_mode & S_IROTH)
            printf("Readable by Others\n");
        else
            printf("Not readable by others\n");
        if(buf.st_mode & S_IXOTH)
            printf("Executable by Others\n");
        else
            printf("Not executable by others\n");
    }

    return 0;
}
[s2019103562@centos8-linux Mon Mar 15 10:13 AM ~]$

```

Stat64() is for large file support.

8.ERRNO():

```

[s2019103562@centos8-linux Mon Mar 15 10:21 AM lab05]$ gcc errorno.c -o errorno
[s2019103562@centos8-linux Mon Mar 15 10:21 AM lab05]$ ./errorno
The error generated is 9
Error description:Bad file descriptor
[s2019103562@centos8-linux Mon Mar 15 10:21 AM lab05]$ cat errorno.c
#include<stdio.h>
#include<stdlib.h>
#include<errno.h>
#include<string.h>
#include<unistd.h>
int main(){
    int err;
    if(close(-1)==-1){
        err=errno;
        printf("The error generated is %d\n",err);
        printf("Error description:%s\n",strerror(err));
    }
    return 0;
}
[s2019103562@centos8-linux Mon Mar 15 10:21 AM lab05]$

```

```
[s2019103562@centos8-linux Mon Mar 15 10:44 AM ~]$ vim err.c
[s2019103562@centos8-linux Mon Mar 15 10:46 AM ~]$ gcc err.c -o err
[s2019103562@centos8-linux Mon Mar 15 10:46 AM ~]$ ./err
The error generated: 2
The description of the error: No such file or directory
Error message printed by perror: : No such file or directory
[s2019103562@centos8-linux Mon Mar 15 10:46 AM ~]$ cat err.c
#include<stdio.h>
#include<stdlib.h>
#include<errno.h>
#include<string.h>
#include<unistd.h>
#include<dirent.h>
int main(){
    DIR* dp;
    dp=opendir("lab06");
    if(dp==NULL){
        printf("The error generated: %d\n",errno);
        printf("The description of the error: %s\n",strerror(errno));
        perror("Error message printed by perror: ");
        exit(EXIT_FAILURE);
    }
    else{
        closedir(dp);
        exit(EXIT_SUCCESS);
        printf("Successfully closed!");
    }
    return 0;
}
```

```
[s2019103562@centos8-linux Mon Mar 15 10:46 AM ~]$ █
```

9.FTW():

```
[s2019103562@centos8-linux Mon Mar 15 10:22 AM ~]$ vim ftw.c
[s2019103562@centos8-linux Mon Mar 15 10:26 AM ~]$ gcc ftw.c -o ftw
[s2019103562@centos8-linux Mon Mar 15 10:26 AM ~]$ ./ftw
The object is a directory
[s2019103562@centos8-linux Mon Mar 15 10:26 AM ~]$ ./ftw rewinddir.c
The object is a file

[s2019103562@centos8-linux Mon Mar 15 10:27 AM ~]$ ./ftw lab01
The object is a directory
[s2019103562@centos8-linux Mon Mar 15 10:28 AM ~]$ █
```

```
[s2019103562@centos8-linux Mon Mar 15 10:27 AM ~]$ cat ftw.c
#include<stdio.h>
#include<stdlib.h>
#include<sys/stat.h>
#include<ftw.h>
int list(const char* name,const struct stat* status,int type){
    if(type==FTW_NS){
        printf("Error occurred\n");
        return 0;
    }
    if(type==FTW_F)
        printf("The object is a file\n");
    if(type==FTW_D)
        printf("The object is a directory\n");
    if(type==FTW_DNR)
        printf("The object is a directry that can't be read\n");
}
int main(int argc,char* argv[]){
    if(argc==1)
        ftw(".",list,1);
    else
        ftw(argv[1],list,1);
    return 0;
}
```

10.DIRCNTL():

```
[s2019103562@centos8-linux Mon Mar 15 10:51 AM ~]$ cat dircntl.c
#include<stdio.h>
#include<stdlib.h>
#include<dirent.h>
#define D_GETFLAG 1
#define D_SETFLAG 2
#define D_FLAG_FILTER 0x00000001
#define D_FLAG_STAT 0x00000002
int dircntl(DIR* dir,int cmd, ...);
int main(int argc,char* argv[]){
    DIR* dp;
    int ret;
    dp=opendir("lab02");
    if(dp==NULL)
        printf("Can't open the directory\n");
    else{
        if((ret=dircntl(dp,D_GETFLAG))==1){
            printf("Error occured\n");
            return 0;
        }
        if(ret & D_FLAG_FILTER)
            printf("Directory names are filtered\n");
        else
            printf("Directory names are not filtered\n");
        if(ret & D_FLAG_STAT)
            printf("Servers are asked for extra stat information\n");
        else
            printf("Servers are not asked for extra stat information\n");
    }
    closedir(dp);
    return 0;
}
```

NOTE: Dircntl() is not supported in Linux

```
[s2019103562@centos8-linux Mon Mar 15 10:55 AM ~]$ gcc dircntl.c -o dircntl
/tmp/ccOG08ZP.o: In function `main':
dircntl.c:(.text+0x42): undefined reference to `dircntl'
collect2: error: ld returned 1 exit status
[s2019103562@centos8-linux Mon Mar 15 10:55 AM ~]$
```
