3주차 3차시 디렉터리 다루기

【학습목표】

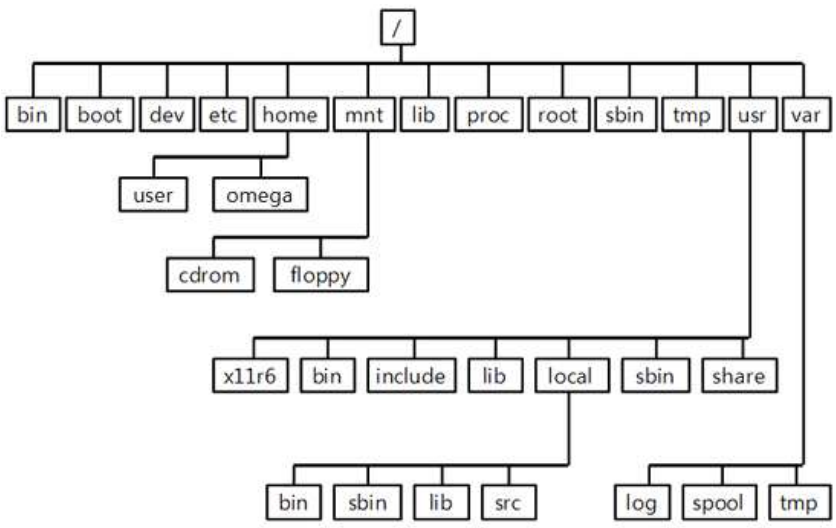
1. 리눅스 디렉터리의 특징을 설명할 수 있다.
2. 디렉터리 생성, 삭제, 수정 등을 할 수 있다.

학습내용1 : 리눅스 디렉터리

1. 디렉터리

① 디렉터리란?
파일의 목록을 저장하기 위한 특수한 형태의 파일
디렉터리 데이터 블록
자기 자신의 항
부모 디렉터리 가르키는 항
자신에게 포함된 파일 목록
아이노드 블록의 번호

② 디렉터리 계층 구조
트리 구조



2. 절대 경로와 상대 경로

* 경로란
파일의 위치
절대 경로
루트 디렉터리를 기준으로 파일의 위치 표현
상대 경로
현재 디렉터리를 기준으로 파일 위치 표현
현재 디렉터리 : .
부모 디렉터리 : ..
디렉터리 구분 : / (윈도우 운영체제는 ₩ 혹은 ＼)

학습내용2 : 리눅스 디렉터리 관련 함수

1. 디렉터리 생성과 삭제

① 디렉터리 생성 : mkdir(2)
path에 지정한 디렉터리를 mode 권한에 따라 생성한다.

```
#include <sys/types.h>
#include <sys/stat.h>
int mkdir(const char *path, mode_t mode);
```

path : 디렉터리 포한된 경로
mode : 접근 권한

② 디렉터리 삭제 : rmdir(2)
성공 시 0, 실패하면 -1를 리턴

```
#include <unistd.h>
int rmdir(const char *path);
```

path : 삭제할 경로

2. 디렉터리 관리

① 디렉터리 이름 변경 : rename(2)

```
#include <stdio.h>
int rename(const char *old, const char *new);
```

old : 변경할 파일/디렉터리 명
new : 새 파일/디렉터리 명

```
01  #include <sys/stat.h>
02  #include <unistd.h>
03  #include <stdlib.h>
04  #include <stdio.h>
05
06  int main(void) {
07      if (mkdir("han", 0755) == -1) {
08          perror("han");
09          exit(1);
10      }
11
12      if (mkdir("bit", 0755) == -1) {
13          perror("bit");
14          exit(1);
15      }
16
17      if (rename("han", "hanbit") == -1) {
18          perror("hanbit");
19          exit(1);
20  }
21
22      if (rmdir("bit") == -1) {
23          perror("bit");
24          exit(1);
25      }
26
27      return 0;
28  }
```

han -> hanbit로 변경

bit는 생성했다 삭제

```
# ex3_13.out
# ls -l
drwxr-xr-x   2 root other 512  1월 12일   18:06 hanbit
```

② 현재 작업 디렉터리 위치 : getcwd(3)
디렉터리 작업 위치 알아낼 때 사용
- 현재 작업 위치 명령 : pwd
실패 시 : NULL 리턴
- 버퍼 주소가 NULL이면 getcwd()는 직업 mallco으로 메모리 할당하고 주소 리턴.

```
#include <unistd.h>
char *getcwd(char *buf, size_t size);
```

buf : 현재 디렉터리의 절대 경로를 저장할 버퍼 주소
size : 버퍼의 크기

③ 디렉터리 이동 : chdir(2)
프로그램에서 디렉터리 이동시 사용
- 리눅스 명령어 : cd

```
#include <unistd.h>
int chdir(const char *path);
```

path : 이동하려는 디렉터리 경로

```
01  #include <unistd.h>
02  #include <stdio.h>
03
04  int main(void) {
05      char *cwd;
06      char wd[BUFSIZ];
07
08      cwd = getcwd(NULL, BUFSIZ);
09      printf("1.Current Directory : %s\n", cwd);
10
11      chdir("hanbit");
12
13      getcwd(wd, BUFSIZ);
14      printf("2.Current Directory : %s\n", wd);
15
16      return 0;
17  }
```

```
# ex3_14.out
1.Current Directory : /export/home/jw/syspro/ch3
2.Current Directory : /export/home/jw/syspro/ch3/hanbit
```

3. 디렉터리 정보 검색

① 디렉터리 열기 : opendir(3)
지정한 디렉터리를 읽기 전용으로 염
성공 시 dir 포인터 리턴, 실패 시 NULL 리턴

```
#include <sys/types.h>
#include <dirent.h>
DIR *opendir(const char *dirname);
```

dirname : 열려는 디렉터리 명

② 디렉터리 닫기 : closedir(3)
지정한 DIR 포인터가 가르키는 디렉터리 닫음
성공 시 0, 실패 시 -1 리턴

```
#include <sys/types.h>
#include <dirent.h>
int closedir(DIR *dirp);
```

dirp : 닫으려는 디렉터리를 가르키는 포인터

③ 디렉터리 정보 읽기 : readdir(3)
지정한 DIR 포인터가 가르키는 디렉터리 내용을 한번에 하나씩 읽음.
성공 시 디렉터리 내용을 하나씩 읽음, 실패 시 NULL을 리턴

```
#include <sys/types.h>
#include <dirent.h>
struct dirent *readdir(DIR *dirp);
```

```
typedef struct dirent {
    ino_t           d_ino;
    off_t           d_off;
    unsigned short  d_reclen;
    char            d_name[1];
} dirent_t;
```

dirp : 정보를 읽어올 디렉터리를 가르키는 포인터
dirent.h 구조체는 sys/dirent.h에 정의

```
01  #include <dirent.h>
02  #include <stdlib.h>
03  #include <stdio.h>
04
05  int main(void) {
06      DIR *dp;
07      struct dirent *dent;
08
09      if ((dp = opendir("hanbit")) == NULL) {
10          perror("opendir: hanbit");
11          exit(1);
12      }
13
14      while ((dent = readdir(dp))) {
15          printf("Name : %s  ", dent->d_name);
16          printf("Inode : %d\n", (int)dent->d_ino);
17      }
18
19      closedir(dp);
20
21      return 0;
22  }
```

```
# ex3_15.out
Name : .  Inode : 208
Name : ..  Inode : 189
```

④ 디렉터리 오프셋 : telldir(3), seekdir(3), rewinddir(3)

파일 오프셋 / 디렉터리 오프셋

디렉터리 오프셋

- 디렉터리 열고, 정보 읽을시 이동
- telldir : 디렉터리 오프셋의 현재 위치를 알려준다.
- seekdir : 디렉터리 오프셋을 loc에 지정한 위치로 이동시킨다.
- rewinddir : 디렉터리 오프셋을 디렉터리의 시작인 0으로 이동시킨다.

```
#include <dirent.h>
long telldir(DIR *dirp);
void seekdir(DIR *dirp, long loc);
void rewinddir(DIR *dirp);
```

dirp : 대상 DIR 포인터

loc : 이동할 위치

```c
#include <sys/stat.h>
#include <dirent.h>
#include <stdlib.h>
#include <stdio.h>

 int main(void) {
     DIR *dp;
     struct dirent *dent;

     if ((dp = opendir("hanbit")) == NULL) {
         perror("opendir");
         exit(1);
    }
     printf("** Directory content **\n");
     printf("Start Offset : %ld\n", telldir(dp));
     while ((dent = readdir(dp))) {
         printf("Read : %s  ", dent->d_name);
         printf("Cur Offset : %ld\n", telldir(dp));
    }
     printf("** Directory Pointer Rewind **\n");
     rewinddir(dp);
     printf("Cur Offset : %ld\n", telldir(dp));

    printf("** Move Directory Pointer **\n");
    seekdir(dp, 24);
     printf("Cur Offset : %ld\n", telldir(dp));

    dent = readdir(dp);
    printf("Read %s  ", dent->d_name);
    printf("Next Offset : %ld\n", telldir(dp));

    closedir(dp);
    return(0);

}
```

```
# ex3_17.out
** Directory content
Start Offset : 0
Read : .
Cur Offset : 12

Read : ..
Cur Offset : 24

Read : han.c
Cur Offset : 512

**   Directory   Pointer
Rewind **
Cur Offset : 0
**    Move    Directory
Pointer **
Cur Offset : 24
Read han.c
Next Offset : 512
```

【학습정리】

1. 디렉터리
파일의 목록을 저장하기 위한 특수한 형태의 파일, root부터 시작하는 계층 구조

2. 절대 경로
루트 디렉터리를 기준으로 파일의 위치 표현

3. 상대 경로
- 현재 디렉터리를 기준으로 파일 위치 표현
- 현재 디렉터리 : .
- 부모 디렉터리 : ..
- 디렉터리 구분 : / (윈도우 운영체제는 ₩ 혹은 ＼)

4. 디렉터리 관련 함수
- 생성 : mkdir()
- 삭제 : rmdir()
- 이름변경 : rename()
- 현재 작업 위치 : getcwd()
- 이동 : chdir()
- 정보 검색 : opendir()
- 닫기 : closedir()
- 정보 읽기 : readdir()
- 오프셋 : telldir(), seekdir(), rewinddir()