4장. 디렉토리, 파일시스템 및 특수 파일

목포해양대 해양전자통신공학부

4.1 서론

- 디렉토리
  - 파일 이름들의 창고
  - UNIX 디렉토리 : 트리 형태의 계층적 구조
- 파일 시스템
  - 디렉토리와 파일들의 집합
  - 파티션
- 특수파일
  - 주변장치까지 파일 개념 확장
  - 파일접근호출로 접근 가능
    - 장치 구동기(device driver) 코드 활성화

4.2 디렉토리

- 홈디렉토리
- 루트(/)
- 현재 작업디렉토리

4.3 디렉토리의 구현 (1)

- UNIX 디렉토리는 파일에 불과
- 몇가지 차이점
  - creat나 open으로 생성되지 않는다
  - 디렉토리 항(entry)들로 구성
    - 하나의 항은 inode 번호라고 부르는 양의 정수와 파일이름을 저장하는 문자필드들로 구성
    - 과거 고정길이
    - 현재는 가변길이(파일 시스템 의존적)
    - inode 번호는 하나의 파일을 유일하게 식별(한 파일시스템 내에서만)
    - inode에 디스크 기반 자료구조를 포함

4.3 디렉토리의 구현 (2)

- 4.3.1 link와 unlink 재고찰
- 4.3.2 점과 이중점
- 4.3.3 디렉토리와 허가
  - 디렉토리 읽기
    - 디렉토리 내용만 읽기 가능
  - 디렉토리 쓰기
  - 디렉토리 수행
    - 디렉토리로 이동하거나 디렉토리 밑에 있는 파일 읽기 가능

4.4 디렉토리와 프로그래밍(1)

- <dirent.h>
- 디렉토리 생성 및 제거
  - mkdir, rmdir
- 디렉토리 열기 및 닫기
  - opendir, closedir
- 디렉토리 읽기
  - readdir
  - rewinddir
- my_double_ls 예제
- find_entry 예제

4.4 디렉토리와 프로그래밍(2)

- 현재 작업 디렉토리
- chdir
- 현재 작업 디렉토리 이름 찾기
  - getcwd
- 디렉토리 트리의 산책
  - int ftw(const char *path, int (*func)(), int depth);
  - int func(const char *name, const struct stat *sptr, int type);

4.5 파일 시스템

- 전통적 파일 시스템
- sync 와 fsync

4.6 UNIX 장치 파일

- /dev
- 블록 장치와 문자 장치
  - 파일시스템은 블록 장치에서만 존재
  - 블록 장치들은 빠른 접근을 위해 연관된 문자 장치를 갖는다(raw 장치)
  - block device switch table / character device switch table
  - 주 장치번호(major device number)
  - 소 장치번호(minor device number)
- stat 구조 재고찰
- 파일시스템 정보
  - statvfs
  - fstatvfs
- 파일과 디렉토리에 대한 제한
  - pathconf, fpathconf

실습과제

- 본문 예제 프로그램 실행 후 결과 확인(필요한 경우 main() 작성)
  - 4.4.3 my_double_ls [50점]
  - 4.4.3 find_entry [50점]
  - 4.4.6 my_pwd [20점]
  - 4.4.7 디렉토리 트리 산책 예 [50점]
  - 4.6.3 파일시스템 정보 예 [50점]
  - 4.6.4 파일 제한값 출력 예 [30점]
- 종합 실습과제 [200점]
  - my_ls
    - 파일시스템에 대한 정보 출력 (정보 내용은 자유)
    - 현재 디렉토리 위치 출력
    - 일반 파일이면 파일에 대한 정보를 한 줄로 출력(출력할 정보 내용과 형식은 자유)
    - 디렉토리인 경우 하부에 있는 파일 개수 출력(그외 추가 정보 출력도 가능)