

강의자료에서 설명된 함수 정리 (4장~12장)

4장: C 표준 파일 입출력

fopen()

파일 열기.

사용법: FILE *fopen(const char *filename, const char *mode);

모드:

"r": 읽기 전용.

"w": 쓰기 전용.

"a": 추가 쓰기.

fclose()

열린 파일을 닫고 리소스 해제.

fgetc() / fputc()

파일에서 한 문자를 읽거나 쓰기.

fgets() / fputs()

파일에서 한 줄을 읽거나 쓰기.

fprintf() / fscanf()

포맷화된 데이터를 파일에 쓰거나 읽기.

feof()

파일 끝을 확인.

fflush()

버퍼를 강제로 파일에 기록.

5장: 파일 입출력

`read()` / `write()`

파일 디스크립터를 이용한 읽기/쓰기.

사용법:

`ssize_t read(int fd, void *buf, size_t count);`

`ssize_t write(int fd, const void *buf, size_t count);`

`lseek()`

파일 위치 포인터를 이동.

사용법: `off_t lseek(int fd, off_t offset, int whence);`

6장: 파일 시스템

`stat()` / `fstat()` / `lstat()`

파일 상태 정보를 가져옴.

사용법: `int stat(const char *pathname, struct stat *buf);`

`chmod()`

파일 권한 변경.

`unlink()`

파일 삭제.

`rename()`

파일 이름 변경.

`opendir()` / `readdir()` / `closedir()`

디렉터리 열기, 읽기, 닫기.

7장: 파일 및 레코드 잠금

`fcntl()`

파일 제어 및 잠금 기능 제공.

사용법: `int fcntl(int fd, int cmd, ...);`

`lockf()`

파일 잠금.

8장: 프로세스

`fork()`

부모 프로세스를 복제해 자식 프로세스 생성.

`exec()`

현재 프로세스를 새 프로그램으로 대체.

`wait()` / `waitpid()`

자식 프로세스 종료를 기다림.

`getpid()` / `getppid()`

현재 프로세스 또는 부모 프로세스 ID 반환.

9장: 프로세스 제어

`nice()`

프로세스 우선순위 변경.

`kill()`

특정 프로세스에 시그널 전송.

10장: 메모리 관리

`shmget()`

공유 메모리 생성.

사용법: `int shmget(key_t key, size_t size, int shmflg);`

`shmat()` / `shmdt()`

공유 메모리를 프로세스에 연결하거나 분리.

`shmctl()`

공유 메모리 제어.

11장: 시그널

`signal()`

시그널 처리기를 설정.

`raise()`

현재 프로세스에서 시그널 발생.

`kill()`

특정 프로세스에 시그널 전송.

12장: 파이프

pipe()

익명 파이프 생성.

사용법: int pipe(int pipefd[2]);

popen() / pclose()

명령어를 실행하고 그 결과를 읽거나 쓰는 파일 스트림 제공.

사용법:

FILE *popen(const char *command, const char *type);

int pclose(FILE *stream);

dup() / dup2()

파일 디스크립터 복제.

mkfifo()

이름 있는 파이프 생성.