cp - Copy Files

SYNTAX:

```
cp [options] source dest
cp [options] source... directory
```

OPTIONS:

- `cp -a` : Archive files
- `cp -f` : Force copy by removing destination file if needed
- `cp -i` : Interactive - ask before overwrite
- `cp -l` : Link files instead of copying
- `cp -L` : Follow symbolic links
- `cp -n` : No file overwrite
- `cp -R` : Recursive copy (including hidden files)
- `cp -u` : Update - copy when source is newer than destination
- `cp -v` : Verbose - print informative messages

---

ls - List Directory Contents

SYNTAX:

```
ls [options] [file|dir]
```

OPTIONS:

- `ls -a` : List all files including hidden files (starting with '.')
- `ls --color` : Colored list [=always/never/auto]
- `ls -d` : List directories with '*/'
- `ls -F` : Add one character of */=>@| to entries
- `ls -i` : List file's inode index number
- `ls -l` : List with long format - show permissions
- `ls -la` : List long format including hidden files
- `ls -lh` : List long format with readable file size

- `ls -ls` : List with long format with file size
- `ls -r` : List in reverse order
- `ls -R` : List directory tree recursively
- `ls -s` : List file size
- `ls -S` : Sort by file size
- `ls -t` : Sort by time & date
- `ls -X` : Sort by extension name

C Implementation of ls Command

```c
#include <sys/types.h>
#include <sys/dir.h>
#include <sys/param.h>
#include <stdio.h>
#define FALSE 0
#define TRUE 1
extern int alphasort();
char pathname[MAXPATHLEN];

main() {
    int count, i;
    struct dirent **files;
    int file_select();

    if (getwd(pathname) == NULL ) {
        printf("Error getting path\n");
        exit(0);
    }
    printf("Current Working Directory = %s\n", pathname);
    count = scandir(pathname, &files, file_select, alphasort);
    if (count <= 0) {
        printf("No files in this directory\n");
        exit(0);
    }
    printf("Number of files = %d\n", count);
    for (i = 1; i < count; i++)
        printf("%s \n", files[i-1]->d_name);
}

int file_select(struct direct *entry) {
    if ((strcmp(entry->d_name, ".") == 0) || (strcmp(entry->d_name, "..") ==
0))
```

```
        return (FALSE);
    else
        return (TRUE);
}
```

---

mv - Rename/Move Files

SYNTAX:

```
mv [options] source dest
mv [options] source... directory
```

OPTIONS:

- `mv -f` : Force move by overwriting destination file without prompt
- `mv -i` : Interactive prompt before overwrite
- `mv -u` : Update - move when source is newer than destination
- `mv -v` : Verbose - print source and destination files
- `man mv` : Display help manual

C Implementation of mv Command

```
#include<sys/types.h>
#include<sys/stat.h>
#include<stdio.h>
#include<fcntl.h>

main(int argc, char *argv[]) {
    int i, fd1, fd2;
    char *file1, *file2, buf[2];

    file1 = argv[1];
    file2 = argv[2];
    printf("file1=%s file2=%s", file1, file2);

    fd1 = open(file1, O_RDONLY, 0777);
    fd2 = creat(file2, 0777);

    while (i = read(fd1, buf, 1) > 0)
        write(fd2, buf, 1);

    remove(file1);
    close(fd1);
    close(fd2);
}
```