

1. Aufgabe

1. rm:

- (a) löscht Dateien, aber Standardmäßig keine Directories
- (b) /bin/rm
- (c) rm test.txt
- (d) unlink()

2. mv:

- (a) verschiebt Dateien, kann auch zum umbenennen genutzt werden
- (b) /bin/mv
- (c) mv test.txt /Documents/test.txt
- (d) access() und rename()

3. chmod:

- (a) ändert die Zugriffsrechte von Dateien
- (b) /bin/chmod
- (c) symbolisch: chmod +rwx file, binär: chmod 777 file
- (d) pathconf()

4. chown:

- (a) ändert den Besitzer einer Datei
- (b) Debian: /bin/chown, FreeBSD: /usr/bin/chown
- (c) chown [user] file
- (d) munmap(), exit()

5. mkdir:

- (a) erstellt Ordner
- (b) /bin/mkdir
- (c) mkdir name
- (d) sigprocmask()

6. rmdir:

- (a) löscht Ordner
- (b) /bin/rmdir
- (c) rmdir name
- (d) execve(), mmap()

7. kill:

- (a) sendet über die PID Signale an Prozesse
- (b) /bin/kill
- (c) kill [PID]
- (d) lstat()

8. ln:

- (a) steht fuer link, erzeugt Verknüpfung zu Datei oder Ordner
- (b) /bin/ln
- (c) ln [option] ZIEL [NameDerVerknüpfung]
- (d) openat(), fstat()

9. sleep:

- (a) laesst den Prozess fuer eine angegebene Zeit warten
- (b) /bin/sleep
- (c) sleep 0.1h
- (d) fstatfs()

10. wget:

- (a) Programm um Dateien aus dem Terminal von ftp- oder http-Servern zu laden
- (b) /usr/local/bin/wget
- (c) wget http://example.com/folder/file
- (d) read(), write()

2. Aufgabe

```
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <time.h>
#include <sys/param.h>

int main (int argc, char *argv[]) {
    struct stat st;

    if (argc < 2) {
        printf("Usage: %s <pathname>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    if (stat(argv[1], &st) == -1) {
        perror(argv[1]);
        exit(EXIT_FAILURE);
    }

    for(int i=1; i < argc; i++){

        lstat(argv[i], &st);
        char type[20];

        if(i > 1){
            printf("\n\n");
        }
    }
}
```

```
    if (S_ISREG(st.st_mode)){
        strcpy(type, "regular file");
    }else if (S_ISLNK(st.st_mode)){
        strcpy(type, "symbolic link");
    }else if (S_ISDIR(st.st_mode)){
        strcpy(type, "directory");
    }else if (S_ISCHR(st.st_mode)){
        strcpy(type, "character device");
    }else if (S_ISFIFO(st.st_mode)){
        strcpy(type, "FIFO (named Pipe)");
    }else if (S_ISSOCK(st.st_mode)){
        strcpy(type, "socket");
    }

    printf("File: \t\t\t %s \n", argv[i]);
    printf("Filetype\t\t %s \n", type);
    printf("UserID:\t\t\t %d \n", st.st_uid);
    printf("GroupID: \t\t %d \n", st.st_gid);
    printf("letzter Zugriff:\t %s", ctime(&st.st_atime));
    printf("letzte Inodeaenderung:\t %s", ctime(&st.st_ctime));
    printf("letzte Aenderung: \t %s", ctime(&st.st_mtime));

    #ifdef __FreeBSD__
    printf("Datei angelegt: \t %s", ctime(&st.st_birthtime));
    #endif

}

exit(EXIT_SUCCESS);
}
```
