

Betriebssysteme im Wintersemester 2021/2022

Aufgabenblatt 7

Mithilfe der Funktion „getopt()“ wurden die Optionen, die eingegeben wurden, ausgelesen.
Mit „opendir()“ wird ein Verzeichnis geöffnet.

```
bool a = false;
bool l = false;
bool g = false;
bool o = false;
char arg = getopt(argc, argv,
                  "algo"); // iteriert über argv, vergleicht die Parameter
// aus argv mit shortopts und gibt die gesuchten Parameter als char zurück
while (arg != -1) { // wenn der getopt() keine neuen Parameter mehr fin-
// det, wird eine -1 zurückgegeben
    switch (arg) {
        case 'a' :
            a = true;
            break;
        case 'l' :
            l = true;
            break;
        case 'g' :
            g = true;
            break;
        case 'o' :
            o = true;
            break;
    }
    arg = getopt(argc, argv, "algo");
}
char *directory = argv[argc -
                      1]; // Zielpfad getopt() tauscht die Reihenfolge von
// argv, Zielpfad liegt jetzt an letzter Stelle
DIR *dir = opendir(directory);
dirent *file = nullptr;
```

Diese Schleife wird so lange ausgeführt, bis es keine Dateien im Verzeichnis mehr gibt.

Die Variable „d_name“ liefert den Namen der Datei.

Mit der Funktion „lstat()“ werden die Metadaten einer Datei in die Variable buf vom Typ „stat“
geschrieben und dann anschließend wird mit der Variable „st_mode“ eine Variable vom Typ
„mode_t“ zurück gegeben.

```
while ((file = readdir(dir)) != nullptr) {
    char *path = file->d_name;
    stringstream ss;
    ss << directory << "/" << path;
    if (*path != '.' || a) {
        struct stat buf;
        if (lstat((char *) ss.str().c_str(), &buf) != 0) // lstat schreibt
```

```

Datei Informationen in das struct buf
    cout << "Errno: " << path << " " << strerror(errno) << endl;
mode_t m = buf.st_mode;
char fileType = '-';
if (l || o || g) {
    if (S_ISDIR(m) != 0) fileType = 'd';
    if (S_ISCHR(m) != 0) fileType = 'c';
    if (S_ISBLK(m) != 0) fileType = 'b';
    if (S_ISFIFO(m) != 0) fileType = 'p';
    if (S_ISSOCK(m) != 0) fileType = 's';
    cout << fileType;
    if (S_IRUSR & m) cout << "r"; // Binärer Und Vergleich
    else cout << "-";
    if (S_IWUSR & m) cout << "w";
    else cout << "-";
    if (S_IXUSR & m) cout << "x";
    else cout << "-";
    if (S_IRGRP & m) cout << "r";
    else cout << "-";
    if (S_IWGRP & m) cout << "w";
    else cout << "-";
    if (S_IXGRP & m) cout << "x";
    else cout << "-";
    if (S_IROTH & m) cout << "r";
    else cout << "-";
    if (S_IWOTH & m) cout << "w";
    else cout << "-";
    if (S_IXOTH & m) cout << "x";
    else cout << "-";

    cout << " " << buf.st_nlink << " ";
    if ((o && !g) || !o && !g) {
        struct passwd *p = nullptr;
        p = getpwuid(buf.st_uid);
        cout << p->pw_name << " ";
    }
    if ((g && !o) || !o && !g) {
        struct group *grp = nullptr;
        grp = getgrgid(buf.st_gid);
        cout << grp->gr_name << " ";
    }
    cout << buf.st_size << " ";
    struct tm *tmptr = nullptr;
    tmptr = localtime(&buf.st_mtim.tv_sec);
    char ptr[80];
    strftime(ptr, 80, "%b %d %X", tmptr);
    cout << ptr << " ";

}
cout << path;
if (l || g || o) cout << endl;
else cout << " ";
}
}
if (!(l || g || o)) cout << endl;

int closeDir = closedir(dir);

```