Patrick Felschen, Matr. Nr. 932056

Julian Voß, Matr. Nr. 934505

Aufgabenblatt 7

Implementation

Eingabe

```
ListDir listDir;
int c;
while ((c = getopt(argc, argv, "aglo")) != -1) {
    switch (c) {
        case 'a':
            listDir.setShowAll(true);
            break;
        //[…]
    }
}
char *directory;
if (optind < argc) {
    directory = argv[optind];
} else {
    directory = ".";
}
listDir.setDir(directory);
listDir.printResult();
```

Für die Eingabe wird mit einer Schleife über die angegebenen Parameter iteriert. Die *getopt* Funktion teilt hierfür die Eingabe in einzelne Parameter auf, auf welche anschließend mittels *switch-case* reagiert werden kann. Je nach Parameter werden im *ListDir* Objekt einzelne *booleans* gesetzt. Bevor der Inhalt ausgegeben werden kann, wird versucht ein Verzeichnis-Pfad auszulesen, wurde kein Pfad angegeben, wird dieser mit dem aktuellen Verzeichnis (.) initialisiert.

Ausgabe

Ein Verzeichnis kann geöffnet werden, indem die Funktion *opendir* mit einem Verzeichnispfad aufgerufen wird. Über den Pointer, der die Funktion *readdir* liefert, kann über den Inhalt des Verzeichnisses iteriert werden. Ein solcher Verzeichniseintrag (*struct dirent*) hält verschiedene Informationen, wie zum Beispiel Dateiname, Zugriffsrechte oder die User-ID. Die Funktion *closedir* beendet das Auslesen des Verzeichnisses.

```
errno = 0;
currentDir = opendir(dir);
if (errno != 0) {
    printf("%s \n", strerror(errno));
    return;
}
```

Bei fehlerhaften Funktionsaufrufen wird ein entsprechender Fehlercode in der globalen Variable *errno* festgehalten. Nach jedem Aufruf von *opendir, readdir* und *closedir* wird überprüft, ob ein Fehlercode in *errno* vorhanden ist. Die Fehlermeldung wird über die Funktion *strerror* auf der Konsole ausgegeben.

Tests

Es wurde getestet, wie das Programm auf nichtexistierende Verzeichnis-Pfade reagiert.

Eingabe: ./myls nichtExistierenderPfad

Aufgabe: No such file or directory