



Systemnahe Software I (WS 2012/2013)

Abgabe bis zum 31. Januar 2013, 16:00 Uhr

Lernziele:

- Verzeichnisse auslesen (`opendir`, `readdir`, `closedir`) und erstellen (`mkdir`)
- Eigenschaften von Dateien auslesen (`stat`, `lstat`) und manipulieren

Aufgabe 16: Verzeichnisse aktualisieren (10 Punkte)

Das hier gesuchte Programm soll die Namen zweier Verzeichnisse als Eingabeparameter erhalten und die im ersten Verzeichnis (= Quellverzeichnis, Q) gefundenen regulären Dateien und Ordner rekursiv in das Zweite (= Zielverzeichnis, Z) kopieren, falls entsprechende Einträge dort noch nicht existieren oder dort auch gegebenenfalls durch neuere Versionen ersetzen, falls schon gleichnamige Dateien vorhanden sind.

Im Einzelnen soll das Programm für jeden Eintrag a in Q Folgendes machen:

- Ist a eine reguläre Datei (`S_ISREG`; siehe Manual-Seite von `stat`) und gibt es keinen gleichnamigen Eintrag in Z , so soll a nach Z kopiert werden. Beim Kopieren ist darauf zu achten, dass
 - ... die Rechte ebenfalls kopiert werden. Die Rechte einer Datei können mit Hilfe von `lstat` ausgelesen und mit `chmod` gesetzt werden.
 - ... der Zeitstempel übernommen wird. Folgende Funktion erledigt dies:

```
bool copy_time_stamp(const char* source_path,
                    const char* target_path) {
    // get stat from source
    struct stat stat_buf;

    if (lstat(source_path, &stat_buf) < 0) {
        perror(source_path);
        return false;
    }
}
```

```

        // copy atime & mtime
        struct timespec times[2];
        times[0] = stat_buf.st_atim; // "last access time"
        times[1] = stat_buf.st_mtim; // "last modification time"
        if (utimensat(AT_FDCWD, target_path, times, 0) < 0) {
            return false;
        }
        return true;
    }
}

```

- Ist *a* eine reguläre Datei in *Q* und gibt es eine gleichnamige reguläre Datei *b* in *Z*, die älter ist, d. h. deren letzte Änderung (= „modification time“) weiter zurückliegt, so soll *b* durch *a* ersetzt werden.

```

...// obtain stat from a (stat_a) and b (stat_b)
if (statb_b.st_mtim.tv_nsec < stat_a.st_mtim.tv_nsec) {
    // b older than a
}

```

- Ist *a* ein Verzeichnis (`S_ISDIR`) und existiert kein gleichnamiges Unterverzeichnis *b* in *Z*, so soll *a* nach *Z* kopiert werden, falls es keinen anderen Eintrag mit gleichem Namen gibt. (Tipp: Nach dem Erstellen des Verzeichnisses *b* mit `mkdir` einfach einen rekursiven Aufruf mit Argumenten *a* und *b* machen.)
- Ist *a* ein Verzeichnis und existiert ein gleichnamiges Verzeichnis *b* in *Z*, so soll rekursiv wie beschrieben mit den Einträgen der neuen Quelle *a* und dem neuen Ziel *b* verfahren werden.
- Bei allen anderen Konstellationen soll nichts gemacht werden.

Hinweise

- Ihr arbeitet auf dem Dateisystem! Wendet also besondere Sorgfalt an: Bitte räumt eventuell entstehende Dateiansammlungen oder Verzeichnisbäume anschließend wieder auf. Euer Programm soll sicher stellen, dass die Tiefe eines neu erzeugten Verzeichnisbaums die Zahl 100 nicht überschreitet, damit im Falle einer Endlos-Rekursion größere Schwierigkeiten vermieden werden können. Testet Euer Programm sorgfältig, bevor Ihr „.“ oder „..“ als Argumente verwendet.
- Beim Auslesen des Verzeichnisses dürft Ihr nicht vergessen, dass „.“ und „..“ ebenfalls Einträge sind, die von `readdir` zurückgeliefert werden!

Den Quellcode-Ordner bitte mittels `tar` packen und dann einreichen:

```

thales$ tar -cvf updatedir.tar updatedir/
thales$ submit ssl 16 updatedir.tar team

```

Viel Erfolg!