

## 11 Quest

### 11.1 Aufgabe

DOS	copy	del	rename	cd	dir	attrib	scandisk	defrag
UNIX	cp	rm	mv	cd	ls	chmod, chown, chgrp	ln	

Anders als unter DOS ist es unter UNIX oft notwendig für bestimmte Befehle root-Rechte zu besitzen, etwa bei dem Befehl `mount`, mit dem man zusätzliche Platten/Hardware einbinden kann.

### 11.2 Aufgabe

#### 11.2.1 FAT (DOS)

File Allocation Table → zentrale Verwaltung

Anfangssektor bekannt, jeder Sektor zeigt auf seinen Folgesektor, Ende durch -1 signalisiert.

Vorteil: gut bei sequentiellen Zugriffen

Nachteil: die Tabelle benötigt (vor allem bei großen Platten) viel Speicher

#### 11.2.2 I-Nodes (UNIX)

Sektortabelle → dezentrale Verwaltung

Für jede geöffnete Datei wird eine Struktur aufgebaut: Attribute, direkte Adressen, einfach/zweifach/dreifach indirekte Adressen

Struktur nur im Speicher, wenn Datei geöffnet wurde

Vorteil: bei random access vorteilhaft, Speicherverbrauch abhängig vom aktuellen Bedarf (offene Dateien), große Dateien möglich (durch indirekte Adressen)

### 11.3 Aufgabe

---

```

int get_block(int offset)
{
    int i;
    struct table f;
    struct entry * e;
    while (i < f.count)
    {
        e = &(f.seq[i]);
        if (offset < e->length)
            return e->address+offset;
        offset -= e->length;
        i++;
    }
}

```

---

### 11.4 Aufgabe

---

```

struct inode
{
    struct stat_attr;
    long d_address[10];
    long indirect1;
    long indirect2;
    long indirect3;
}

struct indirect
{
    long address[100];
}

```

---

Maximal mögliche Dateigröße:

$$(10 + 100 + 100 \cdot 100 + 100 \cdot 100 \cdot 100) \cdot 1 \text{ kB} = 1010110 \text{ kB}$$

### 11.5 Aufgabe

Windows: Name, Größe, erstellt, geändert, letzter Zugriff, einfache Berechtigungen, Attribute (versteckt, Archiv, System etc.)

UNIX: insb. erweiterte Rechte (`ugo +-rwx`)

### 11.6 Aufgabe

```

a.txt  5, 11, FF
b.dll  10, FE
c.exe  0, 4, 2, 14, 8, FF
d.bat  9, FF

```

Konsistenz:

- Kontrolle auf doppelt verwendete Blöcke
- Kontrolle auf Länge der Dateien  
a.txt → 1024 Byte
- Nicht verwendete Blöcke als frei markieren  
1, 3, 6, 7, 12, 15 als frei markieren (FE)
- Kontrolle, ob alle verwendeten Blöcke richtig referenziert wurden  
10 als Ende der Datei markieren (FF)