

Aufgabe 1 (4 Punkte)

Die Performance von Dateioperationen unter Linux sinkt mit der Zahl der (systeminternen) Kopieroperationen und der gewählten Puffergröße. Ein Beispiel dafür zeigt das vorgegebene Programm `mycp`, das ihr in `/home/ti2/ueb/06` als Vorgabe findet. Es dient in sehr naiver Weise dazu, eine Datei zu kopieren.

Beweist, dass ihr es besser könnt! Euer Programm soll wie die Vorgabe zwei Dateinamen als Argument von der Kommandozeile entgegennehmen: Aus der ersten Datei wird gelesen, in die zweite Datei (die ggf. neu angelegt werden muss) wird der Inhalt der ersten Datei geschrieben. Ein Aufruf des Programms könnte z.\ B. lauten:

```
mycp eingabe ausgabe
```

Euer Programm soll beliebig lange Eingabedateien "on the fly" verarbeiten können (d.\ h. Ihr könnt nicht einfach die gesamte Datei in den Speicher laden, sondern müsst diese blockweise bearbeiten). Ein Block soll der Größe einer Page entsprechen, die ihr mit `sysconf(_SC_PAGESIZE)` ermitteln müsst, und zum Einlesen und zum Schreiben mittels `mmap()` in den Hauptspeicher eingebündelt werden (`munmap()` zum Entfernen nicht vergessen!).

Aus euren Tests soll hervorgehen, dass euer Programm korrekt funktioniert und erheblich schneller arbeitet als die naive Referenzimplementierung. Ihr könnt dazu unter anderem das Shell-Kommando `time` einsetzen. Achtet darauf, dass eure Messungen nicht durch Caching von Daten verfälscht werden.¹

Stellt geeignete Testfälle vor. (Kann Euer Programm mit einer leeren Eingabedatei umgehen? Kann Euer Programm mit einer Eingabedatei umgehen, die eine ungerade Länge hat? Kommt es mit Nullbytes in den Nutzdaten zurecht?)

Aufgabe 2 (3 Punkte)

Die Datei `nikolaus.avi` ist 139586400 B groß. Sie ist auf einer Festplatte mit den folgenden technischen Parametern gespeichert:

- 7200 Umdrehungen in der Minute,
- acht Oberflächen zur Datenspeicherung,
- 100000 Spuren pro Oberfläche,
- 1200 Sektoren pro Spur (sei vereinfachend für alle Spuren gleich),
- 512 Bytes pro Plattenblock (Sektor),
- 2.\ ms Kopfumschaltung beim Wechsel der Oberfläche (in einem Zylinder) und
- ein beliebiger Spurwechsel dauert (stark vereinfacht) insgesamt 4.\ ms (dies beinhaltet eine möglicherweise erforderliche Kopfumschaltung).

Wir gehen für Eure Berechnungen davon aus, dass keine weitere Abbildung stattfindet, es sich also um die tatsächliche Geometrie der Festplatte handelt.

a) Wie lange würde es bei bestmöglichen Bedingungen dauern, die genannte Datei von der beschriebenen Festplatte zu lesen und welche durchschnittliche Lesegeschwindigkeit (Datenrate) würde dabei erreicht? b) Wie lange würde es bei der schlechtestmöglichen Verteilung der Datenblöcke auf der Festplatte dauern, bis die genannte Datei von der beschriebenen Festplatte gelesen wurde und welche durchschnittliche Lesegeschwindigkeit (Datenrate) würde dabei erreicht? c) Wie ändert sich das Ergebnis für b), wenn jeweils acht hintereinanderliegende Plattenblöcke zu einem logischen Block der Größe 4 KiB zusammengefasst werden (ein logischer Block wird immer komplett genutzt, selbst wenn nicht alle Plattenblöcke darin benötigt würden, um die Datei zu speichern).

Hinweis: Ihr könnt in eurer Rechnung von den Verwaltungsinformationen der Datei bzw. des Dateisystems abstrahieren. Externe Faktoren wie Stromausfall usw. sollen in der Betrachtung natürlich nicht berücksichtigt werden, ebenso sollen mögliche Optimierungen durch den Festplattencontroller oder das Betriebssystem, wie z.\ B. vorausschauendes Lesen, nicht in die Betrachtung einbezogen werden.

Achtet bei euren Berechnungen darauf, erst bei der Bestimmung des Endergebnisses zu runden. Das Rechnen mit gerundeten Zwischenergebnissen führt zu Punktabzug. Rundet die Datenrate auf KiB/s bzw. MiB/s (beachtet die IEC-Einheiten für Datendurchsatz!)

Aufgabe 3 (3 Punkte)

Warteschlangen/Puffer in Gerätecontrollern sollen dabei helfen, das jeweilige Gerät von der CPU zu entkoppeln. Dies sollt Ihr anhand einer Beispielrechnung illustrieren:

Ein gedachter Gerätecontroller kann ein angeschlossenes Gerät mit einer Datenrate von 10000 Byte/s versorgen. Erreicht er die Low-Watermark, löst er einen Interrupt aus, damit der zuständige Prozess aufgeweckt wird und neue Daten an den Controller gesendet werden. Das Aufwecken des Prozesses samt Nachfüllen der Warteschlange im Controller dauert vereinfachend immer 5\ ms.

Welche effektive Datenrate (in Byte/s) zum angeschlossenen Gerät ergibt sich in folgenden Fällen?

a) Größe des Puffers: 1 Byte. Low-Watermark: 0 Byte. b) Größe des Puffers: 1 KiB. Low-Watermark: 0 Byte. c) Größe des Puffers: 1 KiB. Low-Watermark: 25 Byte. d) Wie müsste die Low-Watermark gewählt werden, damit die Netto-Datenrate des Geräts rechnerisch erreicht wird?

Stellt zusätzlich zu den Ergebnissen Euren Rechenweg dar. (Die meisten Geräte arbeiten natürlich mit höheren Datenraten. Wir bleiben hier aber absichtlich bei greifbaren Zahlen.) Ihr dürft im letzten Schritt der Berechnung mathematisch runden. Das Rechnen mit gerundeten Zwischenergebnissen führt zu Punktabzug. Bedenkt bei euren Berechnungen, dass in der Warteschlange immer mit ganzen Bytes gearbeitet wird!

1. Unter Linux z.\ B. mittels `sudo sh -c 'printf 1 >/proc/sys/vm/drop_caches'.` □