

Hard Disk Power Management 1

Implementierung eines Time-out Mechanismus

In den Rechnern akbp1 und akbp2 ist jeweils eine IBM Travelstar Notebook-Festplatte eingebaut; in Linux erscheint sie als hdc.

Als erster Schritt soll der DDT-Algorithmus (*device-dependent time-out*) implementiert werden:

DDT-Algorithmus

Festplatte in den Standby-Modus setzen, falls

$$t_{la} + t_{be} \leq t$$

Die Variablen bedeuten:

- t : die aktuelle Zeit
- t_{la} : wann wurde zuletzt auf die Festplatte zugegriffen (last_access)
- t_{be} : die break-even-Zeit

Die break-even-Zeit soll über das /proc-Dateisystem konfigurierbar sein (implementiert im IDE-Treiber: `drivers/ide/ide.c` und `ide-proc.c`). Die zu erweiternden Einstellungen befinden sich unter `/proc/ide/hdc/settings`.

Dazu soll der Kernel-Thread `ide_power_task` geschrieben werden, der die Moduswechsel der Festplatte kontrolliert. Dieser Thread wird aufgeweckt (z.B. über einen Mutex), wenn die Festplatte aktiviert oder deaktiviert werden soll oder sich der Modus der Festplatte ändert (z.B. durch einen Reset).

Die Struktur `ide_drive_t` (`include/linux/ide.h`) soll dazu um folgende Informationen erweitert werden:

```
unsigned long break_even_period;
unsigned long last_access;
int power_mode;          /* 0=standby, 1=running */
int new_power_mode;      /* 0=standby, 1=running, 2=changed, 3=no_change */
```

Die Variable `power_mode` stellt den Zustand der Festplatte dar (es wird nur zwischen `standby` und `running` unterschieden); `new_power_mode` ist der neue Zustand bei einem Moduswechsel.

Bei einem Zugriff auf die Festplatte (IDE-Treiber) soll die Variable `last_access` aktualisiert werden. Über einen Timer soll einmal die Sekunde die Funktion `ide_check_for_standby` aufgerufen werden, die die aktuelle Zeit mit (`last_access + break_even_period`) vergleicht und bei einer Überschreitung die Festplatte in den Standby-Modus versetzt (`new_power_mode` auf 0 setzen, `ide_power_task` aufwecken). In manchen Situationen ist der Zustand der Festplatte unbekannt. Dann soll der `ide_power_task` mit dem Modus "changed" aufgeweckt werden und selbst herausfinden, in welchem Modus sich die Festplatte befindet.

Bei regelmäßigen Festplattenzugriffen mit ungefähr gleicher Größe bzw. Dauer (z.B. Audio-player) kann der Algorithmus noch verbessert werden, indem nach einem Datentransfer sofort in den Standby-Modus gewechselt wird (und nicht gewartet wird, bis die break-even-Zeit abgelaufen ist). Der Algorithmus soll dann zum DDT/ES-Algorithmus (*device-dependent time-out with early shutdown*) erweitert werden:

DDT/ES-Algorithmus Festplatte in den Standby-Modus setzen, falls $t_{la} + t_{be} \leq t$ oder $t_{fa} + t_{lb} \leq t \leq t_{fa} + t_{lb} + t_1 \quad \text{und} \quad t_{la} + t_2 \leq t$ Die Variablen bedeuten: t : die aktuelle Zeit t_{lb} : die Länge der letzten aktiven Phase (Dauer eines I/O-Transfers, <code>busy_period</code>) t_{fa} : der Zeitpunkt des ersten Zugriffs in der gerade aktiven Phase (<code>first_access</code>) t_{la} : wann wurde zuletzt auf die Festplatte zugegriffen t_{be} : die break-even-Zeit t_1 : Toleranzwert beim Vergleich des letzten mit dem aktuellen Interval (2 Sekunden) t_2 : kleiner Timeout, um das Ende der aktiven Phase zu erkennen (1 Sekunde)

Als `busy period` wird die Dauer zusammengehörender Festplattenzugriffe verstanden; sie beginnt und endet mit einem Festplattenzugriff.

Dazu muss in der Struktur `ide_drive_t` (`include/linux/ide.h`) noch um folgende Variablen ergänzt werden:

```
unsigned long busy_period;
unsigned long first_access;
```

Die Funktion `ide_check_for_standby` muss nun entsprechend modifiziert werden.

Zum Testen liegen im Verzeichnis `/proj/i4embedded/akbp/harddisk/tests` Programme und Shell-Skripte, die Lese- und Schreibzugriffe durchführen. Als Programm mit periodischen Lesezugriffen liegt unter `/proj/i4embedded/akbp/harddisk/amp` ein einfacher MP3-Player.