Az akkumulátoros üzemidő növelése a Laptop Mode segítségével

A merevlemez gazdaságos használatával meghosszabbíthatjuk hordozható gépünk akkumulátoros üzemidejét.

A hordozható gépek megadják nekünk azt a szabadságot, hogy bárhol, bármilyen feladatunkat elvégezhetjük. Sajnos, ha akkumulátoruk lemerül, az élvezetnek vége. Szerencsére több lehetőségünk van az energiával való takarékoskodásra és az akkumulátoros üzemidő növelésére. Megtehetjük például, hogy csökkentjük a processzor sebességét, tompítjuk a kijelző háttérvilágítását vagy lekapcsoljuk a merevlemezt. Az első két dolog eddig is remekül működött ***Linux*** alatt is, ám a merevlemez lekapcsolása csak nehézkesen ment. Még ha sikerült is megoldani a leállítást, ez az állapot sosem tartott elég sokáig ahhoz, hogy érdemleges energia-megtakarítást eredményezzen. Írásomban a ***Laptop Mode***-ot szeretném ismertetni, a ***Linux*** rendszermag egy nemrég megjelent elemét, amely jól használható megoldást biztosít a merevlemez leállítására. Itt csak a 2.6-os ***Linux*** rendszermaghoz készült változatról lesz szó. A ***Laptop Mode*** a 2.4-es változathoz is létezik, de némileg eltérő kiadásban.

A merevlemezek és az akkumulátoros üzemidő kapcsolata

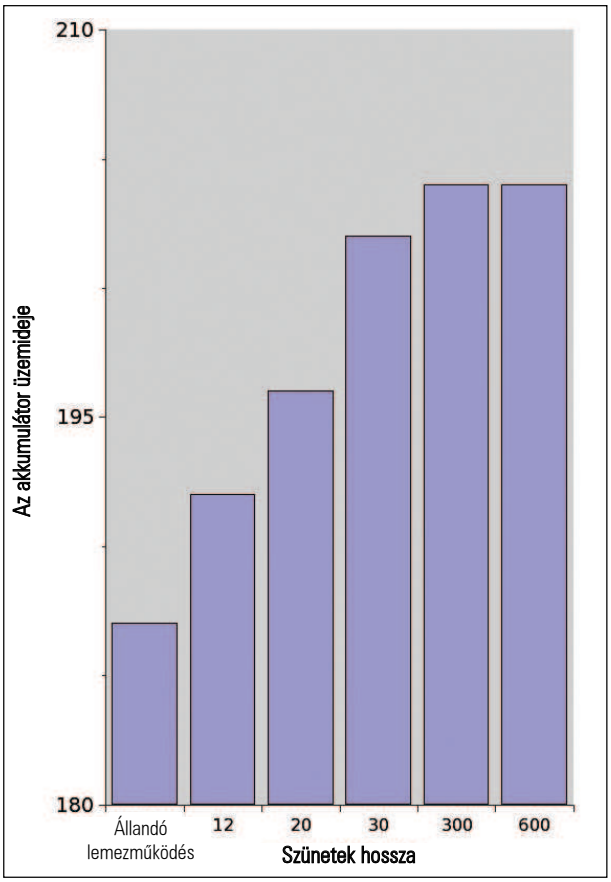
Vajon üzemidő tekintetében mekkora nyereséggel jár a merevlemez lekapcsolása? Próbáljuk meg kiszámítani! Egy átlagos mai hordozható gép lítium-ionos akkumulátora 50–100 wattóra energiát képes tárolni, ami 2-4 óra üzemidőhöz elegendő. Tegyük fel, a saját gépünk akkumulátora 50 wattórás. Ha az akkumulátor a merevlemez bekapcsolt állapota mellett 3,5 órás üzemidőt képes biztosítani, akkor az átlagos energiahasználat 50/3,5 = 14,3 watt. Tegyük fel, hogy a gépben szintén átlagos szintű a merevlemez használata, amely tétlen állapotban 0,9, készenléti módban pedig 0,3 wattot fogyaszt. Az energiahasználatot tehát elméletileg 0,6 watt megtakarításával 13,7 wattra tudjuk mérsékelni. Ezzel az akkumulátoros üzemidő 50/13,7 = 3 órára és 39 percre nő. A nyereség mindig a megtakarított energia és a teljes energiafelvétel viszonyától függ. Esetünkben a leállítással a teljes fogyasztásnak hozzávetőlegesen 4 %-át tudjuk megspórolni, vagyis az üzemidőben is hasonló mértékű növekedésre számíthatunk.

Ennyit az elméletről, nézzünk inkább néhány valódi adatot. Kölcsönkértem egyik barátom ***Apple PowerBook G4*** gépét, feltettem rá a ***Debian GNU/Linuxot*** 2.6.6-os rendszermaggal, és végeztem néhány kísérletet. Arra voltam kíváncsi, hogy legfeljebb mekkora nyereséget lehet elérni az akkumulátoros üzemidőben, és ehhez mekkora időtartamokra kell lekapcsolni a lemezt. Viszonylag jó eredményt vártam, hiszen a gépben egy nagy fogyasztású, 5400-as fordulaton üzemelő lemez volt, és a rendszerből kiszedtem az ***X*** kiszolgálót és az összes démont. Írtam egy teljesítménymérő programot, amely minden órában azonos mennyiségű lemezes be/kiviteli műveletet hajt végre, de úgy, hogy meg lehet adni a be/kiviteli műveletlöketek közötti szünetek hosszát. A tétlen időszakokban a mérőprogram leállítja a merevlemezt. Többféle tétlenségi időhosszal is futtattam a programot, miközben az ***APM*** által szolgáltatott adatok alapján becsültem meg a várható üzemidőt.

A kísérletet folyamatosan működő lemezzel, illetve a löketek közötti idő hosszát 12 másodperc és 10 perc közötti értékekre állítva végeztem el. Eredményeim az *1. ábrán* láthatók. Mint az ábráról is kitűnik, ha 30 másodpercenként átlagosan egynél kevesebb be/kiviteli műveletet végez a gép, már számottevő megtakarítást érhetünk el. Meglepő, hiszen a lemez felpörgetése sok energiát igényel, nem igaz? Nem, valójában nem így van. Ha a lemez két másodperc alatt felpörög, azzal nagyjából annyi energiát vesz fel, amennyit nyolc másodpercig tartó tétlen állapotában használna fel. Ha tehát legalább kilenc másodpercig leállítva tudjuk tartani a lemezt, már nyertünk valamennyit. A löketek közötti 30 másodperces időköz viszonylag hosszú leállást jelent, ezért kaptam ilyen kedvező eredményt.

A Laptop Mode

A ***Laptop Mode*** lényegében a ***Linux*** rendszermag egy beállítása, amely a lemezes be/kiviteli műveletek időbeli elosztását változtatja meg. A ***Linux*** normál esetben kisebb, időben jól elosztott adagokban végzi el a lemezes be/kiviteleket. Ha viszont a lemez, ámbár kényelmesen is, de folyamatosan dolgozik, akkor soha nem tud leállni, és értékes energiát veszítünk. A hordozható gépeknél a lemezműveleteket rövid időszeleteken belül kell végrehajtani, majd ezek között hosszabb szünetet kell tartani, ahogy az a kísérletemnél is történt. Ha engedélyezzük a ***Laptop Mode-ot***, a ***Linux***



1. ábra Az akkumulátoros üzemidőt vizsgáló kísérlet eredménye

is pontosan ezt teszi. A lemezműveletek között ez esetben akár 10 perc is eltelhet, így komoly növekedést érhetünk el az akkumulátoros üzemidőt tekintve.

Hogyan működik?

Tekintsük át azt, hogy a ***Laptop Mode*** hogyan éri el ezt a szokásostól eltérő be/kiviteli viselkedést. Ha tétlen időszakokat akarunk létrehozni, akkor az aktív időszakaszokban a lehető legtöbb be/kiviteli műveletet végre kell hajtanunk. Ha azt aktív időszak véget ért, akkor a lehető leghosszabb ideig vissza kell tartanunk az írási és az olvasási kérelmeket. Az aktív időszakokban meg kell próbálnunk némi pluszmunkát végezni. Először is, bizonyos mértékű előreolvasást kell végrehajtanunk. Ha az adatokra a tétlen időszakban valóban szükség lesz, akkor megtakarítottunk egy felpörgetést. A ***Laptop Mode*** alapesetben 4 MB-ot olvas előre. Ugyancsak fontos, hogy az aktív időszak végén minden változást írjunk is ki a lemezre. Ezzel biztosítani tudjuk az adatok biztonságát. Ha leáll a lemez, tudjuk, hogy a leállásig elvégzett munkánk már biztonságban van.

A tétlen időszakokban az írás az egyetlen visszatartható művelettípus, feltéve, hogy a kiírandó adatokat tudjuk tárolni a memóriában – ezt is csak addig tehetjük, amíg a memória meg nem telik. Sajnos ezt elmondani könnyebb, mint megvalósítani, a ***Linux*** ugyanis számos különböző helyről indít írásra vonatkozó kéréseket, ezeket aztán mind úgy kell módosítani, hogy lehetővé tegyék az írások visszatartását.

Az első és legfontosabb a módosított, vagyis piszkos adatok kezelése. Normál esetben, ha egy gyorstárazott lemezlap több mint 30 másodperce módosult, akkor elavultnak számít, és a pdflush démon kiírja a lemezre. Szerencsére az elavulási időtartam hossza módosítható (***/proc/sys/vm/dirty_expire_centisecs***). A ***Laptop Mode*** itt tíz perces időtartamot ad meg, vagyis a módosítások akár ennyi ideig is a memóriában maradhatnak, mielőtt a lemezre kerülnének. Mivel minden aktív időszak szinkronizálással végződik, a tétlen időszakok piszkos lapok létezése nélkül kezdődnek. A tétlen időszakok első tíz percében tehát biztosak lehetünk abban, hogy elavulás miatt egyetlen lap sem íródik vissza a lemezre.

A második fontos módosítás a naplózó fájlrendszerekre vonatkozik, hiszen ezek önmagukban is rengeteg lemezműveletet végeznek. A ***Laptop Mode*** által támogatott naplózó fájlrendszerek nagy részénél a fájlrendszerre vonatkozó változások egy öt másodpercen belüli írási műveletet váltanak ki. Az ***ext3***-as fájlrendszernél például a fájlrendszerre vonatkozó tranzakcióknak van egy maximális élettartamuk, ennek lejártával azonnal a lemezre kerülnek, ami értelemszerűen egy írási műveletet jelent. A maximális élettartam a `mount commit` kapcsolójával adható meg. Ha leválasztunk és újra befűzünk egy fájlrendszert úgy, hogy ezt az élettartamot tíz percre állítjuk, akkor az ***ext3*** nem fog tranzakciókat kiírni a tétlen időszakok alatt. Ismétlem, a tétlen időszakokat szinkronizálás előzi meg, vagyis a tétlen időszakok kezdetekor nincsenek nyitva hagyott tranzakciók. A ***Laptop Mode*** hasonló módosításokkal él a ***ReiserFS*** és az ***XFS*** esetében is.

Az utolsó változás a Linux memóriakezelését érinti. Ha tétlen időszakban nagymennyiségű memória lefoglalására kerül sor, a memóriakezelőnek ki kell választania néhány eldobandó lapot. Előfordulhat, hogy olyan lapot választ, amelynek tartalmát eldobás előtt ki kell írni a lemezre, mert például módosított lemezlapról vagy cseretárhelyre írandó lapról van szó. Ilyenkor persze fel kell pörgetni a merevlemezt, márpedig éppen ezt szeretnénk elkerülni. ***Andrew Morton*** úgy módosította a memóriakezelőt, hogy az a ***Laptop Mode*** használatakor először kiírást nem kívánó lapokat próbáljon eldobni. Mindezek a módosítások együttesen azt teszik lehetővé, hogy a ***Laptop Mode*** használatakor akár tíz percig is lekapcsolva maradjon a merevlemez. Ha nem túlságosan gyakran módosítunk fájlokat, akkor ennél is hosszabb időszakokat húzhatunk ki a merevlemez használata nélkül. Végül is, ha nincs mit írni, a meghajtót sem szükséges felébreszteni. Sajnos, ha a fájlrendszereket alapértelmezett beállításokkal fűzzük be, akkor a helyzet megváltozik, a fájlrendszer ugyanis rögzíti a hozzáférési időket. A hozzáférési idők akkor is frissítésre kerülnek, ha csak olvassuk a fájlokat, ilyenkor azonban ki kell írni azokat a lemezre. Ezt a kellemetlenséget elkerülendő a ***Laptop Mode*** leválasztja, majd a `mount noatime` kapcsolóját használva fűzi be újra a fájlrendszereket. Ezzel megszűnik a hozzáférési idők rögzítése, és lehetővé válik, hogy hosszabb időtartamot is képesek legyünk eltölteni lemezes be/kivitel nélkül.

Talán már feltűnt, hogy az itt művelt dolgokat – mint a ***/proc*** piszkálása – jellemzően felhasználói térből szokás elvégezni. Nem véletlen, hogy a ***Laptop Mode*** is egy rendszermag összetevőből és egy felhasználói térbeli parancsfájlból áll. A ***Laptop Mode-ot a*** parancsfájl segítségével lehet

Tippek és csapdák

MP3-lejátszás

Ha úgy akarunk MP3 fájlokat lejátszani, hogy a merevlemez kikapcsolt állapotban marad, próbáljuk meg növelni a */sbin/laptop_mode READAHEAD* beállításánál megadott értéket. Az is jó megoldás, hogy az összes MP3 fájlt egy kisebb tmpfs RAM-lemezre másoljuk. Ügyeljünk arra, hogy ne legyen túl nagy a RAM-lemez, mert akkor a gép a cseretárhelyre fogja kiírni, és eredeti szándékunkkal pontosan ellentétes eredményt érünk el. Kisméretű RAM-lemezen sok dolgot lehet és érdemes tárolni, ha kikapcsolt állapotban akarjuk tartani a merevlemezt, de természetesen csak akkor, ha az adatok esetleges elvesztése nem okoz gondot.

Egyedi leállási idők

A *Laptop Mode* maximális leállási idejét módosítani is lehet. Ehhez a */sbin/laptop_mode* fájlban szereplő MAX_AGE beállításnak kell azt az értéket adni, ahány másodpercig lemezre íratlanul, a memóriában akarjuk tartani az adatainkat. Az alapérték 600 másodperc, vagyis tíz perc. Amint a *PowerBook-kal* végzett kísérletem eredményéből is látszik, különösebben nem éri meg ennél magasabb értéket megadni. Ha ezt az időt csökkentjük, akkor kevésbé kell aggódnunk adataink elvesztése miatt, de veszítünk nagyjából két perc üzemidőt. Ha valamilyen valóban fontos adatot szeretnénk biztonságban tudni, akkor annak mentése után magunk is elindíthatunk egy szinkronizálást. A *Laptop Mode* figyelembe veszi az ilyen kéréseket, és ilyenkor mindent kiír a lemezre. A szinkronizálás alapállapotba hozza az időzítőket is, vagyis elvégzése után a lemez akár újabb tíz percig is leállítva maradhat.

Leállítás a cpudyn segítségével

Ha cpudynt használunk a processzor órajelének dinamikus szabályozására, talán érdekel bennünket, hogy a program a merevlemez leállítására is képes. Vannak olyan meghajtók, amelyek a tétlenségi időtúllépés beállítását egyszerűen figyelmen kívül hagyják, ha túl alacsonynak találják a kapott értéket. Ráadásul a tétlenségi idő értékét öt másodperces lépésekben lehet beállítani, ezért például nyolc másodpercet nem tudunk megadni. A cpudyn itt jön a képbe, ugyanis a merevlemez közreműködése nélkül figyeli a tétlen időszakokat, és akkor állítja le a merevlemezt, amikor csak akarjuk.

Smart Spindown

A *Laptop Mode* az aktív időszakok végén végez egy szinkronizálást, majd megvárja, hogy a lemez önmagától leálljon. Ugyanakkor jobb, ha a szinkronizálást közvetlenül a meghajtó leállása előtt hajtjuk végre, így ugyanis nagyjából 20 másodperccel frissebb adatokat tudunk kiírni. A *Laptop Mode* erre nem képes, ugyanis nem tud aktív lemezlekapcsolást végezni. *Smart Spindown* névvel írtam egy parancsfájlt, amely képes együttműködni a *Laptop Mode*-dal. Szépnek vagy tökéletesnek véletlenül sem mondanám, de ha minden cseppnyi energiával takarékoskodni akarunk, esetleg az adatok biztonsága szempontjából számít az a húsz másodperc, akkor legalább van mihez nyúlnunk. (Lásd az internetes forrásokat.)

engedélyezni, ez a rendszermag oldali támogatást a */proc/sys/vm/laptop_mode* beállításával kapcsolja be. Ezután leválasztja és újra befűzi a fájlrendszereket, miközben a */proc* bizonyos beállításait is módosítja.

Üzembe helyezés

A ***Laptop Mode*** használatához először is szükségünk van egy a támogatására képes rendszermagra. A ***Laptop Mode a*** 2.6-os ***Linuxokban a*** 2.6.6-os változattól felfelé található meg. A rendszermag forrásfájában a ***Laptop Mode*** leírását a ***Documentation/laptop-mode.txt*** fájlban találjuk. A leírásba beágyazva szerepel a vezérlő parancsfájl, ezt magunknak kell kimásolnunk és elmentenünk `/sbin/laptop_mode` név alatt. Ezután adjunk a parancsfájlra futtatási jogot: `chmod 700 /sbin/laptop_mode`.

A ***Laptop Mode*** engedélyezéséhez rootként a `/sbin/laptop_mode start` parancsot kell kiadni. A parancsfájl minden szükséges műveletet elvégez, kivéve a merevlemez leállítását, amelyhez a meghajtó tétlenségi időtúllépését is meg kell adnunk – erre a `hdparm -S 4 /dev/hda` parancs szolgál. A 4-es érték 20 másodperces tétlenségi időtúllépésre utal. Ha le akarjuk tiltani a ***Laptop Mode-ot***, csak adjuk ki a `/sbin/laptop_mode stop` parancsot.

Célszerű lehet a ***Laptop Mode-ot*** úgy beállítani, hogy akkumulátoros üzemmódra váltáskor azonnal elinduljon. Ha ACPI-támogatással rendelkező gépünk van, akkor tegyük a következőket: másoljuk ki az ***ac_adapter*** és a ***battery.sh*** fájlt a ***Laptop Mode*** leírásából, majd helyezzük őket a megadott helyre. Szerkesszük át a ***battery.sh***-t, adjuk meg benne merevlemezünk eszköznevét és a kívánt tétlenségi időt – és ennyi az egész.

Rejtélyes bekapcsolások

Időnként a merevlemez úgy pörög fel, hogy ennek semmi okát nem látjuk. Ilyenkor meg kell keresni a jelenség hátterét. A ***Laptop Mode*** ismer egy úgynevezett blokk-kiíratómódot is, amelyet a lemezhasználat hibakeresésére lehet alkalmazni. Mielőtt engedélyeznénk, tiltsuk le a ***syslogd***-ben a rendszermag üzeneteinek naplózását, esetleg állítsuk is le teljesen. Ennek módja terjesztésünk felépítésétől függ. Ha nem állítjuk le a ***syslogd***-t, gépünk végtelen ciklusba eshet, ugyanis a hibakeresés kimenetét átveszi a ***syslogd***, majd kiírja lemezre, ám ezzel újabb merevlemezes művelet naplózását váltja ki és így tovább.

A blokk-kiíratómód engedélyezéséhez rootként az `echo 1 > /proc/sys/vm/block_dump` parancsot kell kiadnunk. A rendszermag kimenetében, amelyet most, hogy a ***syslogd*** nem működik, a `dmesg` segítségével olvashatunk el, az alábbiakhoz hasonló üzeneteket fogunk látni:

```
bash(273): READ block 3242 on hda1 (A 3242-es
 ➥blokk olvasása a hda1 eszközről)
bash(273): dirtied inode 10237 (.bash_history) on
 ➥hda1 (A hda1 eszköz 10237-es fájlleírója
 ➥bepiszkolódott)
pdflush(6): WRITE block 3242 on hda1 (írás a hda1
 ➥eszköz 3242-es blokkjába)
```

A kimenet értelmezése a következő. Egy ***bash*** nevű, 273-as azonosítójú folyamat kiolvasta a */dev/hda1* eszköz 3242-es blokkját. Ugyanez a folyamat bepiszkolta a ***.bash_history*** nevű fájlt. A fájl tehát megváltozott, de még nem íródott ki a lemezre. A ***pdflush*** démon ezután írta a 3242-es blokkot, ez alighanem a ***bash*** által korábban módosítottal egyezik meg.

Ha kezünkben a hibakeresési kimenet, megkereshetjük a hiba forrását. Ha *READ* (olvasás) üzenetet látunk, akkor célhoz is értünk. Ki kell találnunk, hogy az adott folyamatnak miért volt szüksége az adatokra, majd eldönthetjük, hogy leállítjuk a folyamatot, vagy megváltoztatjuk az alkalmazás beállításait úgy, hogy többé ne igényelje az adatokat, esetleg olvassa be őket előre, amikor a merevlemez éppen működik. Ha egy fájlt szeretnénk előreolvasni, nem kell mást tennünk, mint hogy kiadjuk a `cat /konyvtar/fajl >/dev/null` parancsot, lehetőleg kétszer is, így a *Linux* csak olvasható alrendszere nem dobja el azonnal a fájlt. Ha csak bepiszkolt fájlra utaló üzenetet látunk, akkor nem kell aggódnunk. Senki és semmi nem ír a lemezre, ezek az üzenetek csak azt jelzik, hogy egy folyamat alkalomadtán kiírást igénylő módosításokat hajt végre. Ilyenkor a merevlemez csak tíz percenként pörög fel, majd rögzíti a változásokat, más nem történik.

Ha tíz percnél gyakrabban látunk *WRITE* üzeneteket, de *READ* üzeneteket nem találunk, vagyis nincs olyan művelet, ami aktiválná a merevlemezt, akkor valószínűleg valamelyik folyamat közvetlenül szinkronizálja az általa kezelt fájlok tartalmát. A *syslogd* például hajlamos ilyesmire. Ha azt látjuk, hogy a *syslogd* nem várt pillanatokban végez írásokat, akkor módosítanunk kell a *syslog.conf* tartalmát. Valószínűleg van benne egy ilyen sor:

```
kern.*      /var/log/kern.log
```

Ez arra utasítja a *syslogd*-t, hogy minden a *kern** maszkkal egyezést mutató naplóüzenet után indítson el egy fsync() hívást. Ha a sort a következőképpen változtatjuk meg:

```
kern.*      -/var/log/kern.log
```

majd újraindítjuk a *syslogd*-t, akkor ezeknél az üzeneteknél többé nem fog `fsync()` hívásokat indítani. Ügyeljünk viszont arra, hogy mely naplófájlokra vonatkozóan végzünk módosításokat. Ha figyelünk a biztonságra, akkor például fontosnak tarthatjuk az *auth.log* szinkronban tartását.

Köszönetnyilvánítás

Ugyan nekem nincs hordozható gépem, mégis rengeteg örömet szerzett nekem a *Laptop Mode-on* végzett munka. Szeretnék köszönetet mondani mindazoknak, akik hozzájárultak a *Laptop Mode* fejlesztéséhez, köztük *Jens Axboenak, Micha Feiginnek, Andrew Mortonnak* és *Kiko Pirisnek*. Ugyancsak hálás vagyok *Jeroen Kruis* segítségéért, aki hagyta, hogy vadonatúj *PowerBook G4*-esét használjam a kísérleteimhez.

Linux Journal 2004. szeptember, 125. szám

Bart Samwel a Laptop Mode az őkezdeményezése nyomán lett a 2.6-os Linux része – noha ő maga nem rendelkezik hordozható géppel. Informatikát tanul a Leideni Egyetemen, Hollandiában, miközben egyedi programok írásából él meg. A ➲ bart@samwel.tk címen érhető el.