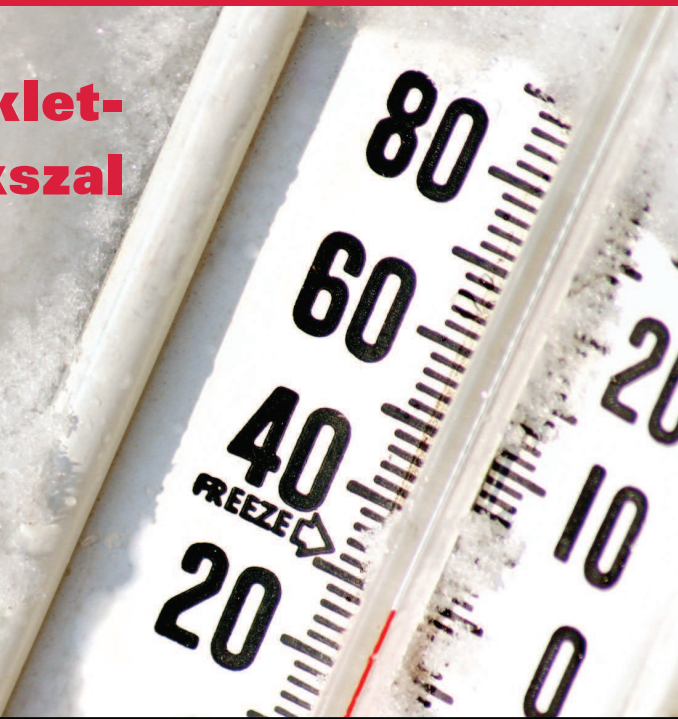


Távoli hőmérséklet-figyelés a Linuxszal

Készítsünk távoli hőmérséklet-figyelő berendezést kis igényű Linux és olcsó vas felhasználásával.



■ Egy hűtés, fűtés és légkezelés területén tevékenykedő barátommal folytatott beszélgetésem után vágtam bele ebbe a projektbe a hőmérsékleti adatok távolról történő eléréséhez és rögzítéséhez. Az ő feladata biztosítani, hogy a kinti időjárástól függően kellemes legyen a benti légkör – nem túl meleg, de nem is túl hideg. Tapasztalata szerint sok új berendezésben adódnak kezdeti hibák, amelyeket meg kell oldani, hiszen nincs két pontosan egyforma összeállítás. A munka utolsó szakasza a legnyomasztóbb. A befejezés után a vásárló felhívja, és közli, hogy valami nincs rendben. A megrendelő elégedetlen, de a barátom nem tudja, hol kezdje keresni a problémát, mert nincs semmi használható objektív információ a berendezés hibás működésének mibenlétéről.

Egyetértettünk abban, hogy hasznos lenne rögzíteni a külső hőmérsékletet, és elektronikusan naplózni a méréseket. Ez lenne a hibakeresési folyamat javításának egyik módja. Ezután elkezdtem olcsó, könnyen telepíthető és egyszerűen használható készre gyártott rögzítőeszközt keresni. A keresgélés során rengeteg kereskedelmi terméket és készletet találtam. Voltak önállóak, és voltak olyanok, amelyek számítógép segítségével jelenítették meg és rögzítették a hőmérsékleti ada-

tokat. A három követelmény egyformán fontos volt, és a termékek többségét túl drágának találtam a költségvetéshez képest. A telepítés egyszerűsége volt a másik jellemző probléma. Bizonyos eszközök huzalozása bonyolult volt, vagy követelmény volt, hogy a hőmérséklet mérésének helyén kell elhelyezni. Sokan úgy gondolják, hogy a hőfokszabályzó csak elcsúfítja a falat, ezért nem lehetett volna könnyen meggyőzni őket, hogy még egy dobozt szereljenek fel a hőmérséklet rögzítésére.

A linuxos megoldás

A költségek és a telepítés problémáinak orvoslására végül megvizsgáltam a rendszer alkotóelemként történő felépítésének lehetőségét. A *Linux* elvét követve, amely az eszközök összegyűjtésével és egyesítésével oldja meg a feladatokat, arra gondoltam, hogy egy digitális multimétert és egy számítógépet használok, valamint egy programot, amely segítségével együtt tudnak működni. A terv szerint az *RS-232* soros kapuval ellátott digitális multiméter egy érzékelővel méri a hőmérsékletet. A számítógép összegyűjti a multiméter adatait, és feldolgozza azokat a megjelenítéshez. Ismertem *Linuxot* használó nyílt forráskódú alkalmazásokat a multiméterhez, és korábban vásároltam egy

multimétert általános hibakereséshez. Rendelkezésre állt egy leselejtezett *PC*, így minden szükséges összetevő a kezemben volt a hőmérséklet-rögzítő rendszer mintapéldányának megépítéséhez.

A digitális multiméterhez tartozott egy soros kapukábel, valamint egy *DOS* program. A mellékelt *DOS* programot nem használtam, mert nem lehetett úgy módosítani, hogy lehetővé váljon az érzékelővel történő hőmérséklet-mérés. Ehelyett teljesen új programot írtam a *QuickBasic* segítségével. Rendelkezésre álltak a multiméter soros kapujára vonatkozó szükséges információk, a *QuickBasicben* pedig minden olyan szolgáltatás adott volt, amelyekre szükségem volt. Sikerült működésbe hoznom egy kezdeti kommunikációs programot, de a program méretének növekedésével a *DOS* és a *QuickBasic* memóriakezelési problémáiba ütköztem, különösen amikor hozzáláttam megoldani az adatok megjelenítésének és rögzítésének igényét. Ezen a ponton nagy visszalépésnek tűnt a memóriakezeléssel küzdeni. Tudtam, hogy a *Linux* olyan környezetet biztosítana, ahol nem kell a memóriakezeléssel foglalkoznom, ezért egy olyan terjesztés után néztem, amellyel helyettesíthetem a *DOS*-t. Kiderült, hogy a *Linux* legnépszerűbb

terjesztései nem megfelelőek ehhez az alkalmazáshoz. Még a terjesztések minimális telepítése is meghaladná a leselejtezett gép teljesítményét. Találtam egy terjesztést, amely felül-emelkedett a korlátokon: a *University Linuxot*, *Paul Mullertől*, amely kis memória- és lemezigénnyel rendelkezik. Kevesebb, mint **20 MB DOS** által formatált lemezterület és **24 MB RAM** felhasználásával tudtam futtatni. A legnagyobb előny az, hogy ez a terjesztés jól tűri az áramszüneteket. Ha kimarad az energiaellátás, a számítógép anélkül indul újra, hogy kézi beavatkozást igénylő fájlkárosodást okozna. Ez pénzt takarít meg, és csökkenti a rendszer összetettségét, mert nem kell szünetmentes áramforrást (*UPS*) alkalmazni ahhoz, hogy a rendszer az áramkimaradások idején tovább működjön. Miután mindent beállítottam a gépen, mér nem volt szükség billentyűzetre vagy monitorra. Egy *Windows PC* és *Telnet*, valamint *Ethernet* kapcsolat segítségével tudtam kommunikálni a rendszerszámítógéppel a fejlesztéshez és teszteléshez. A folyamatos írás és tesztelés híve vagyok, ezért a *Perl* nyelvet választottam a projekthez. A *University Linuxban* a *Perl 5.003* változata található meg. Nem tudtam *Perl* modulokat használni, mert az alkalmazás mérete túl kicsi volt, és ez kisebb kényelmetlenséget jelentett. A *University Linux* az *Acme Labs thttpd* kiszolgálóját is tartalmazza. Ennek segítségével úgy lehet beállítani a rendszert, hogy egy webböngésző segítségével jelenítse meg a hőmérséklet-méréseket.

A hardver tesztelése

Tandy 22-805 típusszámú digitális multimétert használtam, amelyhez egy használati utasítás, egy *DOS* program, mérővezetékek és egy 9 tűcsatlakozóval ellátott kábel tartozik. A kézikönyv szerint az adatátviteli beállítások 600 baud, 7 adatbit, 2 stop bit és nincs paritás. Egy fontos adatot kihagytak a leírásból, de a szükséges információt megtaláltam az interneten. A *DTR* és az *RTS* vezetékre külön figyelmet kell fordítani. A *DTR* vezetéket alacsonyra, az *RTS* vezetéket pedig magasra kell állítani ahhoz, hogy a multiméter adatátvitelt tudjon folytatni a soros kapun keresztül. Ha a két

vezeték ettől eltérően van beállítva, lehetetlen adatot kapni a multimétertől. Ennél a terjesztésnél csak az *stty*-t tudtam soros kommunikációra használni, és a parancsfájlból nem lehetett közvetlenül vezérelni a *DTR* és az *RTS* vezetéket.

Azt tapasztaltam, hogy amikor a parancsfájlból meghívom a *stty*-t, a *DTR* és az *RTS* alacsonyból magasba vált. Ez a *DTR* esetében jó, de az *RTS*-nek alacsonyan kell maradnia. Észrevettem, hogy a számítógép második soros kapuján alacsony az *RTS*, és nincs használatban. Ha a multiméter *RTS* soros felület kábelét a második soros kapu *RTS* tűjéhez csatlakoztatnám, elhithetném a multiméterrel, hogy a helyes vezetékeállításokat látja. Egyszerűen eltávolítottam az *RTS* vezetéket a multiméterből, és a második soros portra kötöttem. Miután megoldottam ezt a problémát, bekapcsoltam a multimétert és az alábbi rövid próba-parancsfájlt (*serialtest.pl*) hoztam össze:

```
# !/usr/bin/perl
#
# serialtest.pl
#
# Parancsfájl 22-805 típusú
# Tandy multiméter
# soros kapun keresztül történő
# olvasáshoz
$port = "/dev/ttyS1"; # set to
↳ COM1
system ("stty 600 cs7 cstopb
↳ clocal -ixon -echo < $port");
open (SERIALPORT, "+>$port") or
↳ die "can't open $port. ";
print SERIALPORT ("\n"); # take
# a reading
$R = <SERIALPORT>; # read
# returned string
print "$R" ;
close (SERIALPORT); # close
# port
exit 0;
```

A parancsfájl sikeres futása esetén olyan karakterláncot kell kapnom, amely azonos a multiméter kijelzőjén megjelenő értékkel. Az eredmény:

```
OH O.L MOhm
```

Kezdetnek jó! Ezután a multiméterrel le kellett olvasnom a hőmérséklet-érzékelőt.

Egy *NTC (negative temperature coefficient – negatív hőmérsékleti együtthatójú)* termisztor választottam a hőmérséklet-érzékelőbe. Az ijesztően hangzó név ellenére ez csupán egy kétvezetékes elektronikus alkatrész, amely a hőmérséklettől függően változtatja az elektromos ellenállását. A multiméter esetében az ellenállás érték szolgáltatja a hőmérséklet meghatározó információt. A termisztor nem lehet fordítva bekötni, mert nem érzékeny a feszültségpolarításra (+ vagy -). Ez azt jelenti, hogy a helyszínen a telepítést végző szakembernek eggyel kevesebb dolog miatt kell aggódnia.

A termisztor nem törekszik, de a testéhez csatlakozó vezetékek erős rángatás vagy hajlítás hatására megszakadhatnak. Egy kétállású elosztóblokkot használtam a probléma megoldására és a vezetékekhez történő csatlakozás egyszerűsítésére. Az egyik termisztor kábelt és egy vezetéket egy csavaros lezáró alá helyeztem, majd meghúztam a csavart, hogy biztos mechanikus és elektronikus érintkezést kapjak.

A termisztor a mérővezetékek végeihez csatlakoztatva, a mérővezetékeket pedig a multiméterbe dugva ismét bekapcsoltam a készüléket, majd elindítottam a próba-parancsfájlt. A következő ellenállás értéket kaptam:

```
OH 34.23kOhm
```

Az állás számérték része 34,23, utána pedig egy k betű áll. A k a kilo, azaz 1000 rövidítése. Mivel a multiméter kijelzőjén nincs elég hely a nagy számok megjelenítéséhez, szorzót alkalmaz. Ebben az esetben a 34,23k 34 230 Ohm-ot jelent.

A gyártó által mellékelte ellenállás-hőmérséklet táblázatból kiderítettem, hogy ez az érték nagyon közel van a 0 °C-hoz. Ez megegyezik egy másik hőmérő mérésével, amelynek az érzékelője a közelben volt elhelyezve, ezért biztos voltam abban, hogy az összeállított rendszer működik majd, és pontos értéket szolgáltat.

Most létre kellett hoznom egy parancsfájlt, amely az adatokat felhasználva megjeleníti a hőmérséklet-értéket.

Szoftverválasztás

Kétféleképpen lehetett megoldani a ellenállás-hőmérséklet átváltást a parancsfájlból. Használhattam volna egy adattáblázatot, tömbben tárolva az ellenállás-hőmérséklet értékpárokat. Ennél a megközelítésnél a tömbben található elemek teljes száma jelent hátrányt. Egy -40°C-tól +40°C-ig terjedő tartományban 81 (ne feledjük a 0°C-ot) értékpárra van szükség. Nem létezik olyan egyszerű módszer, amellyel módosítani lehetne termisztor gyártója által rendelkezésre bocsátott szövegfájlt, ha pedig magunk gépeljük be az értékeket, az sok időbe telik, és hibákat eredményezhet.

Ehelyett az úgynevezett *Steinhart-Hart* egyenletet alkalmaztam (lásd a keretes írást). Ezt az egyenletet arra fejlesztették ki a '60-as évek végén, hogy segítse a termisztorokkal gyűjtött óceán-hőmérséklet adatok feldolgozását, és az alkalmazásával közvetlenül hőmérsékletre lehet váltani az ellenállást. Egy interneten talált táblázatkezelő segítségemre volt az egyes termisztorcsaládokra jellemző együtthatók kiszámításában, és ezt használtam az egyenletben.

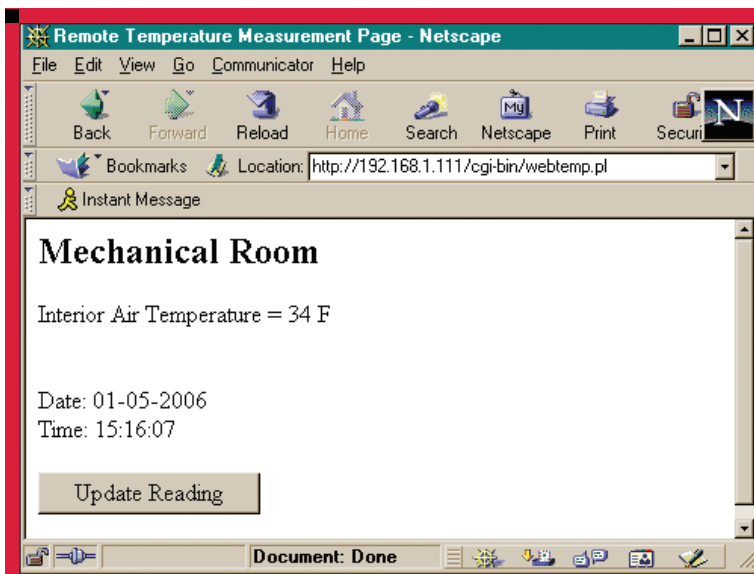
Adatok megjelenítése

Miután a parancsfájl a multiméter méréséből kiszámítja a hőmérsékletet, meg kell jeleníteni, valamint tárolni kell azt. Ezt szem előtt tartva bővítettem a parancsfájlt, hogy átváltsa és megjelenítse a hőmérsékletet, továbbá kiírja az időt és az ellenállás értéket. A *University Linux* 2.0 rendszer magot használ, és engedélyezi a rendszergazda felhasználó bejelentkezését a *Telneten* keresztül. Amikor hagyományos felhasználók próbálják futtatni a *grabtemp.pl* parancsfájlt, hibajelzés jelenik meg a */dev/ttyS1* soros kapunál használt jogosultságok miatt. Ezt a jogosultságok megváltoztatásával küszöböltem ki:

```
chmod a+x /dev/ttyS1
```

Ezután a hagyományos felhasználók is be tudtak jelentkezni, és a parancsfájlt futtatva ellenőrizni tudták a hőmérsékletet. Nem volt szükség rendszergazda hozzáférésre.

Íme az eredményül kapott *showtemp.pl* parancsfájl kimenete:



1. ábra A hőmérséklet-figyelő már webes felülettel rendelkezik

```
/perlserial: perl -w
↳ showtemp.pl
01-05-2006 14:43 34 F 1.3 C
↳ 30.52 k ohms
```

Itt láthatjuk a dátumot, a °F-ben és °C-ban megadott hőmérsékleteket, valamint a pillanatnyi ellenállásértéket. Ellenőriztem a hőmérsékletet ott, ahol az érzékelőt elhelyeztük, és azt láttam, hogy az érték pontos, tehát a parancsfájl átváltási képlete működött. Nem sok számítógép-felhasználó szereti a parancssoros programfelületeket. Az egérrel irányított kezelőfelület sokkal barátságosabb. Tehát ismét kibővítettem a parancsfájlt, hogy lehetővé tegyem a felhasználók számára a rendszer webböngészőn keresztül történő működtetését.

Az üzembe állított és működő *thttpd* kiszolgáló mellett nem volt más dolgom, mint átírányítani a parancsfájl kimenetét úgy, hogy egy weboldalt hozzon létre a megjelenítéshez. Ahogy az alábbi kód mutatja, ez meglehetősen egyszerű:

```
print "content-type: text/html
↳ \n\n";
print "<HTML><BODY><P>";
print "<HEAD><title>Remote
↳ Temperature Measurement
Page</title></HEAD>";
print "<H2>Mechanical
↳ Room</H2> ";
print '<form action=
```

```
↳ "webtemp.pl" method=post>
↳ <P> <P>';
print "Interior Air
↳ Temperature = $out_tempF<BR>";
print "<BR>";
print "<BR>";
print "Date: $out_date <BR>";
print "Time: $out_time <BR>";
print "<BR>";
print '<input type=submit
↳ value="Update Reading">';
print "</form>";
print "</BODY></HTML>";
```

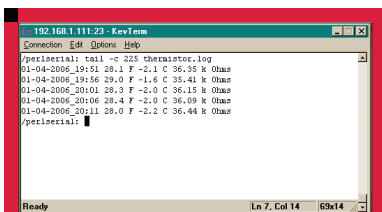
A *webtemp.pl/cgi-bin*-ből történő futtatásakor az 1. ábra példaképernyőjéhez hasonló jelenik meg a felhasználók számára.

Ez a példa a szoba hőmérsékletét, illetve a mérés időpontját és dátumát is megjeleníti. Az *Update Reading* (Leolvasás frissítése) gombra kattintva ismét futtathatjuk a parancsfájlt, és újabb hőmérséklet értéket jeleníthetünk meg.

Nem nehéz olyan kiterjesztést írni a parancsfájllhoz, amely folyamatosan naplózza a hőmérsékletet. Beillesztettem egy sort az *rc* (indító) parancsfájlba, amely elindít egy adatrögzítő parancsfájlt, amely ezután folyamatosan fut a háttérben. Úgy ítélem meg, hogy 5-10 perces időközönkénti méréseket érdemes használni, mivel zárt, légkondicionált térben lassan változik a levegő hőmérséklete.

A termisztorok és a Steinhart-Hart egyenlet

Egy termisztor ellenállás-hőmérséklet függvénye grafikonon ábrázolva nagyon hasonlít egy sí rámpa ívéhez, és minden termisztorcsalád saját, egyedi görbével rendelkezik. Egy egyszerű $y = mx + b$ képlet tehát nem segít az ellenállás hőmérsékletre váltásában. A görbék egyenlete egy többtagú kifejezéssel írható le. A *Steinhart-Hart* egyenlet egy trinomiális, vagyis egy három kifejezésből álló egyenlet. Ahhoz, hogy az egyes ellenállás-méréseknél megoldjuk az egyenletet, három együtthatóra van szükség: az *a*-ra, a *b*-re és a *c*-re. Vannak gyártók, amelyek megadják a termisztorok együtthatóit. Mások csak az ellenállás-hőmérséklet átváltási táblázatot mellékelik. Ha az együtthatók nem állnak rendelkezésre, egy létező táblázatkezelő alkalmazás segíthet kiszámítani azokat a gyártók táblázatai alapján. Nincs szükség számolásra, csak egyszerűen begépeljük a három hőmérséklet és ellenállás értéket a táblázatba, ami automatikusan kiszámítja az együtthatókat. Ezek általában nagyon kis számok, amelyeket tudományos jelölésrendszerrel ábrázolnak. A kiszámított együtthatókat a táblázatból a *Perl* parancsfájlból pásztázva azonban csökkenthetjük a gépelési hibák esélyét.



2. ábra A hőmérséklet-figyelő naplóban tárolt példaértékek így mutatnak a Telneten keresztül

A *Telnet* segítségével a parancssoron keresztül elérhetjük a hőmérséklet naplót. Mivel a formátum helykorlátos, a *Microsoft Excel* segítségével grafikonokat és megjelenítési irányvonalakat készítettem a dátumfájlból. A 2. ábrán egy lehetséges kimenetet láthatunk.

Biztonsági kérdések

A végső cél a hőmérséklet-adatok megjelenítésére és rögzítésére szolgáló, megbízható, könnyen alkalmazható elektronikus eszköz létrehozása volt. A rendszer tényleges kialakítása akkor az elhelyezkedés és a hálózati kapcsolat meglehetősen változó lehet. Minden összeállításnál a körülménye-

ket figyelembe véve kell mérlegelnünk ma biztonsági kérdéseket. Előfordulhat, hogy kerülőutakat kell alkalmaznunk a biztonsági problémák kezelésére. Megtehetjük például, hogy nem a *cgi* könyvtárban található parancsfájl, hanem egy háttérben futó parancsfájl alkalmazásával rögzítjük a hőmérséklet-értékeket szöveg vagy *HTML* formátumban, ezáltal elkülönítjük a naplózás folyamatát a webhozzáféréstől. Másik lehetőségként az egyik kiszolgálóról *FTP*-n vagy *HTTP*-n keresztül gyűjthetjük az adatokat, egy másik biztonságos kiszolgáló segítségével. Ez újabb védőréteget alkot, amely megakadályozza a közvetlen hozzáférést a külvilág felől, mégis elérhetővé teszi az adatokat.

További hivatkozások

A digitális multiméterek általános célú mérőeszközök. Ugyan ennél az összeállításnál termisztor használtam a hőmérséklet mérésre, de bármilyen érzékelőt alkalmazhatunk, amelynek kimenete ellenállás, feszültség vagy elektromos áram. Áramlást, nyomást, súlyt, fényerőt és páratartalmat is mérhetünk.

Nincs szükség több multiméterre, ha több hőmérsékletet akarunk mérni. Megtehetjük, hogy egyetlen multimétert kapcsolóberendezésre kötünk. Ezután egy parancsfájl készítésével a kapcsolóberendezés működtetéséhez, ami lehetővé teszi, hogy egyenként válasszuk ki a hőmérséklet-érzékelőket.

Összegzés

Ez a példa szemlélteti, hogy a *Linux* eszközrelvűsége hogyan működik az olyan alkalmazások megoldásánál, amelyeknél fontos a költség és a rugalmasság. A rendelkezésre álló terjesztések más operációs rendszerekhez képest bő választéka célszerűvé tette a szükséges szolgáltatásokat megvalósító rendszer fejlesztését. Ezen felül a *Perl* és a *University Linux* terjesztés által biztosított fejlesztőfelület segítségével további szolgáltatásokat adhatunk hozzá. A rendszer kevesebb, mint 100 dollárból elkészíthető. A multiméter, a termisztor és a huzalozás tartozékai számos elektronikai cikket forgalmazó kereskedésben elérhetőek. Sok forgalmazó rendelkezik webhellyel, így megrendelés előtt könnyű összehasonlítani a szolgáltatásokat, a tulajdonságokat és az árakat. Használt multiméter vásárlásánál körültekintőnek kell lenni, mert nehéz megállapítani, hogy a korábbi használat befolyásolta-e a műszer pontosságát.

Linux Journal 2006., 144. szám

Steven M. Lapinskas a szoftverminőség-biztosítás, a mechanikus tervezés és a projektvezetés terén rendelkezik szakmai háttérrel. Szabadideje egy részében a *Linux* és a számítógépen kívüli való világ kapcsolatát kutatja.

KAPCSOLÓDÓ CÍMEK

www.linuxjournal.com/article/8833

Garantáljuk weboldalad
100%-os rendelkezésre állását.
 Egyetlen leállás egy hónapban, és visszafizetjük a pénzed.

www.syrius-software.hu