

Szolgáltatára! – munkaütemezés Linux alatt

Egyszerű parancssoros eszköz késleltetett programvégrehajtás tervezésére és kezelésére: az at.

A kapcsolattartás világából kölcsönözve a szolgáltatásokat két nagy csoportra bonthatjuk: azonnaliakra és késleltetettre. Az azonnali kapcsolattartási szolgáltatásokhoz tartozik a telekonferencia, a hálózati csevegés stb., késleltetett szolgáltatás a levél (e-mail) és a fax. Ezek a sorok a Linux at eszközének használatáról adnak rövid áttekintést, amely a programok késleltetett (vagy időzített) végrehajtására képes.

Természetesen a legtöbb programot azonnal szeretnénk indítani. Az sem ritka azonban, hogy kívánatosabb (esetleg egyenesen elkerülhetetlen) lenne, ha a program (illetve munka – job) indítása valamilyen okból kifolyólag egy későbbi időpontban történne meg. Gondoljunk végig a következő eseteket:

1. Egy fejlesztőcsoport kifejlesztett egy parancsfájlt, amely újraépíti a forrásfát és frissíti az alkalmazást. A parancsfájl teljes mértékben gépesített, de igencsak idő- és processzor-igényes. Gyakorlati okokból a csoport úgy döntött, hogy kipróbálásának feladatát egy késő éjszakai időpontra időzíti, amikor viszonylag kevés felhasználóval és folyamattal kell versenyeznie.
2. Egy nagyfelbontású színes nyomtató igen nagy terhelés alatt áll az üzleti órákban, ezért a projektvezető a gyorsan bővülő projektterv több mint 200 oldalát egy héten többször is újra nyomtatja, elkerülve azonban a nyomtató csúcsidőszakbeli teljes kisajátítását, a feladatot a kora reggeli órákra időzíti (még a legtöbb felhasználó megérkezése előtt).
3. A projektvezető levélben emlékeztetőt, illetve napirendet szeretne szétküldeni minden résztvevőnek, négy órával a következő péntek délután három órára időzített találkozó előtt.
4. A rendszergazda jelentést kap, miszerint a munkanap elején a rendszerteljesítmény különlegesen gyenge. A gyengülő teljesítmény oka azonban nem teljesen nyilvánvaló. A rendszergazda a számos naplófájl végigböngészése helyett úgy dönt, időzít egy programot, amely pillanatképet készít a rendszerterhelésről. A feladatot következő nap kilenc órától tízig minden öt percben le kell futtatni.
5. A helyi raktár adatai frissítésére használt külső adatforrás minden péntek délután hat órakor válik elérhetővé. Ha a hosszadalmas frissítési folyamat megszakad vagy leáll, egy órával később próbálkozhat újra, máskülönben a következő péntekre halasztódik el.

A fenti példákban felvázolt esetek időzítési követelményei az at eszköz igénylését tükrözik. Sőt, ezek a példahelyzetek a leggyakoribb at-alkalmazások egészen széles kínálatát ölelik fel. Az at eszményi a nagy számításigényű programoknak kis felhasználói terhelésű időszakokban történő végrehajtására; a szűk erőforrásoknak általában a jobban elérhető időszakokban történő használatára; emlékeztetőknak az adott időpontban való elküldésére; folyamatok olyan időben történő végrehajtására, amikor a felhasználó nem szándékozik aktívan a rendszerhez csatlakozni; illetve olyan munkafolyamatok végrehajtására, amelyek csak valamilyen adott időpontban elérhető erőforrástól függenek.

Az at képességei

Az at tulajdonképpen az e feladatokhoz kapcsolódó programok gyűjteménye, amelyek segítségével a Linux-felhasználók az időzített munkák végrehajtását kezelhetik és ütemezhetik. Azt gondolhatnánk, hogy az at csak olyan folyamatokhoz használható, amelyek nem igényelnek interaktivitást – és ez így is van. Az at segítségével ütemezett programok csak olyanok lehetnek, amelyek háttérfolyamatként is képesek futni, az at ugyanis a végrehajtáshoz nem rendel felhasználói megjelenítő felületet. A felhasználók a parancssoros felületet általában elég rugalmasnak találják, emellett hatékony is, így minden elképzelhető időzítési követelménynek megfelel. Mielőtt belekezdenénk a parancssoros példák bemutatásába, foglaljuk össze az at által nyújtott alapvető szolgáltatásokat. Egyfelől lehetőséget nyújt, hogy a munka végrehajtását egy adott időpontra, illetve dátumra időzítsük, másfelől képes a jelenleg időzített munkákat megjeleníteni, egy adott időzített munkát leállítani, valamint az at használatára jogosult felhasználók listáját kezelni.

Egyéb „at”-tények

- Az at nem cron! Azok a felhasználók, akik az at-et egyáltalán nem vagy csak kevésbé ismerik, hajlamosak egy kalap alá venni a cron-alkalmazásokkal. Igaz mind az at, mind a cron ütemezett feladat-végrehajtást tesz lehetővé, a cron azonban kifejezetten szakaszosan végrehajtott feladatokhoz (például mindennap délután négykor futtatandó programok) lett kifejlesztve. Az at-et ezzel szemben olyankor használjuk, amikor a feladatot csak egyszer szeretnénk – valamilyen adott időben – végrehajtani, vagy amikor a végrehajtás dátumát programozottan kell meghatározni.
- Ha az at más megvalósításait is használtuk már, megfigyelhetjük, hogy a kimenet és a parancssori írásmód tekintetében különbségek vannak a linuxos at és más üzleti Unixok között.
- Mennyire lehet egy feladatot „a jövőbe” időzíteni? A példák többségének futtatásához használt rendszer hamar morcosává vált, amikor a feladatot 2037 utánra időzítettem. Ugyanakkor, nem hiszem, hogy ez bármely felhasználónak komoly hátrányt jelentene.
- Az at és atq programok által megjelenített időértékek olvashatóbbak, ha a POSIXLY_CORRECT változó be van állítva, például: `POSIXLY_CORRECT=1`; `export POSIXLY_CORRECT`
- Az atq parancs a -l kapcsoló, az atrm pedig a -d kapcsoló rokon értelmű kifejezése.

Az „at” használata

Az itt bemutatott példák feltételezik, hogy valamilyen szintű ismeretekkel már bírunk a héjprogramok végrehajtását és a Bourne-héj formai követelményeit illetően. Minden példát kipróbáltam a RedHat Linux 7 (2.3 rendszer) alatt, és úgy

1. táblázat Lehetséges idő- és dátumbejegyzések

Jelölés	Értelmezés
10:15 pm 10/10/2001	2001. október 10. este negyed tizenegy
12:15 7/4/2001	2001. július 4. negyed egy
14:00 7/4/2001	2001. július 4. délután kettő
8 am Apr 15, 2001	a idén április 15. reggel nyolc
10:10 Sep 5, 2001	2001. szeptember 5. tíz óra tízkor
noon	ma (vagy holnap) délben
10 pm tomorrow	holnap este tízkor
teaidő	adott nap (vagy azt követő nap) délután négykor
now + 5 minutes	pontosan öt perc múlva
now + 1 week	pontosan egy hét múlva
tomorrow + 1 hour	pontosan 25 óra múlva

gondolom, ezek minden általános terjesztés alatt változatlan formában működnek.

Az `at` legnagyobb erőssége, hogy segítségével a programokat későbbi időpontra időzíthetjük. Általában a `/usr/bin` könyvtárban találjuk, és alapvető parancssoros írásmódja a következő:

```
at [kapcsol k] ID [D`TUM] < bash-højprogram
```

A fenti írásmód követelményeinek megfelelően a következő hívási módok is ugyanilyen jól működnek:

```
cat bash-højprogram | at [kapcsol k] ID [D`TUM]
```

```
at [kapcsol k] ID [D`TUM]
## BÉrmi, amit rendes esetben a Bourne-høj
## elfogad 0s 0rtelmezni tud. LezÆrva a
## CTRL-D karakterrel
utas tÆs 1
utas tÆs 2
utas tÆs 3
utas tÆs 4
CTRL-D
```

Az első lényeges részlet, amit érdemes megismernünk, az idő és a dátum helyes formázása. A dátum mindig elhagyható. Ha kihagyjuk a parancsból, a program azt a dátumot feltételezi, amihez képest az adott időpont a legközelebb van: azaz vagy az adott, vagy az azt követő napot. Ha a megadott időpont délután fél kettő, de már fél öt van, a feladat másnap fél kettőre lesz időzítve. Az `at` a kiterjesztett POSIX.2 szabványnak megfelelő dátum- és időadatokat beolvasására képes. A jelölésmód leírása valószínűleg megtalálható rendszerünk `/usr/doc` könyvtárában, a `timespec` fájlban.

Az első táblában található idő- és dátummegadási példák elég jól mutatják, mit fogad el az `at` parancs – amint láthatjuk, egy igen gazdag parancshalmazt. A parancssorban kapott értékeket balról jobbra értelmezi. Ha az idő, illetve dátum szabályai sérülnek, a *Garbled time* hibaüzenet jelenik meg, és a program leáll. Általában valamilyen velős magyarázattal is szolgál a zavar okát illetően. Figyeljük meg például a következő hívás-próbálkozást:

```
$ at 6am Mar 32
```

```
Error in day of month. Last token seen: 32
Garbled time
$
```

A következő hívás viszont sikeresen időzíti a feladatot. Emlékezzünk a korábban bemutatott harmadik példaalkalmazásra:

```
$ at 1pm friday
warning: commands will be executed
↳using /bin/sh
    STAFF="moe larry curly"
    cd mail
    mail -s "Meeting Reminder"
    ↳$STAFF < friday_agenda.txt
    CTRL-D
job 6 at Fri Apr 13 13:00:00 2001
$
```

Ha jobban belegondolunk, azért felmerül néhány kérdés. Tárólnak-e valahol a szabványos kimenet- és a hibafolyamok kimenetei? Az `at` szolgáltatás (illetve a `/usr/sbin/atd` program) minden szabványos kimenetet és hibaüzenetet elfog, és amikor az időzített feladat véget ér, levélben elküldi a parancsot kiadó felhasználónak. A levél a következő fejléccel fog megjelenni:

```
Subject: Output from your job 17
```

Vajon csak Bourne-højprogramokat tudunk átadni? Amint a fenti példa visszajelzése is sejteti, az `at` a rendszer Bourne-højprogramját használja (`/bin/sh`) a felhasználó által átadott utasítások értelmezésére. Így bármi, amit a Bourne-højba gépelhetünk, azonnal érvénybe lép, ideértve a Bourne-højprogramokat vagy bármilyen a környezetünkben található végrehajtható állomány indítását, tehát egy másik nyelv értelmezőjét is. Figyeljük meg a következő példát, ahol egy Perl-parancsfájl időzítünk:

```
$ at 6pm tomorrow
warning: commands will be executed
↳using /bin/sh
    perl /home/moe/perls/script1.pl
    CTRL-D
job 28 at Wed Apr 18 18:00:00 2001
$
```

Milyen *felhasználó/csoport* azonosítók rendelhetők az időzített folyamathoz? A munkafolyamat környezetének milyen értékei őrződnek meg és jutnak tovább a végrehajtási környezetbe? Annak feltárása, hogy egy adott eszköz a háttérben miképpen dolgozik, a legtöbb felhasználó számára teljesen közömbös. Az `at` esetében az eszköz működését egy példán szemléltetve egyszerűen kaphatunk képet arról, hogy mit miért tehetünk meg, és mit miért nem. Minden egyes folyamatidőzítés eredményeképpen létrejövő parancsfájl a `/var/spool/at/` könyvtárba kerül. Ezek önműködően létrehozott parancsfájlok és a következő részekre bonthatók:

- Programmegjegyzések, amelyek tájékoztatást adnak a program végrehajtásáról. Ugyanitt található a feladathoz rendelt érvényes felhasználói és csoportazonosítók.
- Az `umask`-beállítások, ezek döntenek a fájlok és könyvtárak létrehozásának mikéntjéről.
- A környezeti változók teljes listája a folyamat elküldésének időpillanatában (kivéve azok, amelyek a megjelenítő eszközökkel kapcsolatosak, mint például a `TERM` és a `DISPLAY`).

2. táblázat Parancssori lehetőségek

Kapcsoló	Viselkedés
-f job_script	A job_script tartalma lesz a feladat tartalma. Ez a kapcsoló akkor lehet érdekes, ha az at parancsot parancsfájlból használjuk.
-m	A feladat elvégzéséről akkor is kap értesítést a feladó felhasználó, ha a program szabályosan lefutott vagy nem lépett fel a hiba.
-v	A felhasználó azonnali idő-, illetve dátumvisszajelzést kap. Azaz, megtudjuk, hogy az adott dátum- és időadatok miképpen értelmeződtek.

3. táblázat Felhasználók szabályozása at-tel

/etc/at.allow	/etc/at.deny	Eredmény
Nem létezik	Nem létezik	Csak a rendszergazdának van jogosultsága.
Létezik	-	Csak a /etc/at.allow fájlban felsoroltak jogosultak.
Nem létezik	Létezik	Mindenki jogosult, kivéve a /etc/at.deny fájlban felsoroltakat. Ez a jellemző alapértelmezett beállítás a Linux telepítésekor.

- A pillanatnyi könyvtár átváltása abba a könyvtárba, ahol a felhasználó a feladat ütemezésekor tartózkodott. Ha a folyamat végrehajtásakor a könyvtár már nem létezik, a folyamat megszakad, és figyelmeztető levelet kapunk.
- Végül az at-nek átadott programszöveg van hozzáfűzve. Ezt mutatja be az 1. lista (23. CD Magazin/at_szolgalatara könyvtár) lerövidített példája, amely a Moe felhasználó által időzített folyamat eredményeképpen létrejött parancsfájl tartalmazza. A program parancssoros felületének kiegészítéséhez vessünk egy pillantást a 2. táblázatra, ahol néhány igen hasznos parancssori lehetőséget találunk. A következő példa a feladat időzítésére mutat még egy módszert:

```
$ at -mf ~/shells/program1 6pm tomorrow
warning: commands will be executed
↳ using /bin/sh
job 29 at Thu Apr 12 18:00:00 2001
$
```

Két másik program is tartozik az at alapprogramjához: az atq és az atrm. Az atq kilistázza a jelenleg időzített feladatokat, míg az atrm egy vagy több feladatidőzítést szakít meg. Míg azonban az összes munkafolyamat megtekintése és megszakítása a rendszergazda jogosultságai közé tartozik, addig a felhasználók csak a saját folyamataikat nézhetik meg, illetve állíthatják le.

A 2. lista (23. CD Magazin/at_szolgalatara könyvtár) az atq által megjelenített kimenetre mutat példát, ahogyan azt a rendszergazda láthatja (a mezőcímkéket az olvashatóság kedvéért én illesztettem be, az atq linuxos változata sajnos nem támogatja őket).

A rank (rang) egy egyedi sorozatszám, amely az időzített feladatok más programok általi azonosítására szolgál. Az időzítési időpontra és a feladó felhasználón kívül a queue (sor) érték található itt. Hacsak a felhasználó másképpen nem rendel-

kezett, a feladatok az a sorba kerülnek. (A különböző queue-értékek használatának mikéntjéről a megfelelő sűgőoldalon olvashatunk – tömören megfogalmazva ez befolyásolja a feladat futásidejű besorolását). Az atq a jelenleg futó feladatokat a queue sorban az = értékkel jelöli, így a 3. táblázatbn látható 17-es feladat éppen az adott pillanatban hajtódik végre. Mi történik, ha a felhasználó elküldött egy feladatot, de ráébredt, hogy az időpont, illetve a program nem a megfelelő volt? Az atrm eszközt egy vagy több időzített feladat eltávolítására használhatjuk. Például a rendszergazda a 3. táblázatban található első és második feladatot a következő parancssal törölheti:

```
atrm 18 19
```

Az atrm nem ad visszajelzést. A parancsot követő atq-listázás pedig a 3. listában (23. CD Magazin/at_szolgalatara könyvtárban) látható eredményt adta.

Az at felügyelete

Ha valamelyik at-program a következő üzenetet adja: *You do not have permission to use at*, lépünk kapcsolatba a rendszergazdával.

Ahogy el is várjuk, a rendszergazda az eszköz használatát illetően teljes körű jogosultsággal rendelkezik, és ugyanezeket a jogokat a felhasználóknak is megadhatja. Két rendszerfájl, a /etc/at.allow és a /etc/at.deny irányítja az at-eszköz elérését. A 3. táblázatban mutatja be, hogy e fájlok megléte és tartalma miképpen befolyásolja az adott rendszeren a felhasználók jogosultságait.

Figyeljünk arra, hogy amennyiben a /etc/at.allow fájl létezik, a rendszer a /etc/at.deny fájlra figyelmen kívül hagyja. A felhasználó mindkét fájlban Linux-azonosítójuk révén vannak megadva, külön-külön, soronként egy névvel. Az at e fájlok szerkesztésére nem ad parancssoros eszközt. Ha szükséges, a rendszergazda ezeket a fájlokat a kedvenc szerkesztőjét használva kézzel szerkeszti.

Összefoglalásul: a felhasználók számára az at használatát közvetlen vagy közvetett módon lehet megtiltani vagy engedni. A rendszergazda a feladatnak megfelelően kezelheti az elérést: kizárásos vagy engedélyezéses alapon. Válasszuk azt a megoldást, amelyik telepítésünknek a legjobban megfelel. Azoknál a gépeknél, amelyeknél a biztonság nagyon fontos, az at-ot engedélyezéses alapon érdemes működtetni (azaz a felhasználóknak nem lesz jogosultságuk, hacsak kifejezetten engedélyt nem kapnak – a /etc/at.allow fájl létezik). Ugyanakkor a Linux alapértelmezett beállítás a legtöbb fejlesztő-, illetve próbakörnyezethez tökéletes lehet (a felhasználók tehát jogosultak, hacsak nincsenek kilitva – a /etc/at.allow nem létezik, és a /etc/at.deny nulla vagy több bejegyzést tartalmaz).

Összegzés

Összességében az at-programok a késleltetett alkalmazás-végrehajtás feladatára kínálnak lehetőséget. Minden egyszerűsége és hasznossága dacára a Linux-rendszergazdák és -fejlesztők gyakran elfeledkeznek az at eszközeiről.



Louis J. Iacona (lji@omnie.com) 1982 óta Linux/Unix alatti alkalmazásokat tervez és fejleszt. Jelenleg az OmniE Labs, Inc vezető embere (☞ <http://www.omnie.com>).