© Kiskapu Kft. Minden jog fenntartva

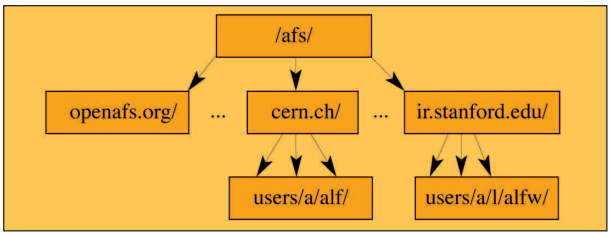
Címtárszolgáltatások (3. rész)
Az AFS – Biztonságos, elosztott fájlrendszer

Egyszeri bejelentkezésre épülő rendszerünket többféle géptípusról is elérhető, elosztott fájlrendszerrel tehetjük teljessé.

Az *Andrew File System (AFS)* egy elosztott, biztonságos, átfogó, az adatok számára helyfüggetlenséget, méretezhetőséget és áttelepítési lehetőséget biztosító fájlrendszer. Az *AFS* sokféle operációs rendszer alól elérhető, és számos nagyobb helyen használják hosszú évek óta. Az *AFS* – közel 20 éves életkora ellenére – egyedi, más elosztott fájlrendszerek által nem biztosított szolgáltatásokat nyújt. Lehet, hogy öregsége miatt egyeseknek nem annyira vonzó, ám miután az *IBM* 2000-ben nyílt forrásúvá tette, használata és fejlesztése új lendületet kapott. Írásomban az *AFS* szolgáltatásait szeretném ismertetni, és szeretnék kedvet csinálni a kipróbálásához.

Az AFS szolgáltatásai és előnyei

Az *AFS* ügyfélprogram *Linux*, illetve a *HP-*, a *Compaq-*, az *IBM-*, a *Sun-* és az *SGI-*féle *UNIX*-változatokhoz, továbbá *Microsoft Windows* és az *Apple Mac OS X*-e alá érhető el. Az *AFS* tehát tökéletes megoldás, ha helyi vagy nagy távolságú hálózaton keresztül többféle operációs rendszer vagy géptípus között szeretnénk adatokat megosztani.
Minden *AFS* ügyfélgép rendelkezik helyi gyorsítótárral. A gyorsítótár-kezelő feladata a gépet igénybe vevő felhasználók követése, valamint a tőlük érkező adatkérések kezelése. Az adatok gyorsítótárazása fájldarabonként történik, a rendszer ezeket az *AFS* fájlkiszolgálóról a helyi lemezre másolja. A gyorsítótárat a gép minden felhasználója igénybe veheti, tartalma az újraindítások során is érintetlen marad. A helyi gyorsítótárazásnak köszönhetően csökken a hálózati forgalom, és a gyorsítótárazott adatok ismételt elérése is felgyorsul.
Az *AFS* egy átfogóan egyedi névtér szerint szerveződik. Az *AFS* fájltér átfogó nézetére láthatunk példát az *1. ábrán*. A fájlokhoz vezető elérési utak neve minden esetben azonos, függetlenül attól, hogy az adatokat honnan érjük el. Az elérési utak nem tartalmaznak kiszolgálóra utaló részt. Mondhatnám úgy is, hogy az *AFS* használója nem tudja, hogy a kívánt adat melyik fájlkiszolgálón található. Ez úgy érhető el, hogy az *AFS* az adatok helyét tároló adatbázist replikálja minden ügyfélre. A megoldás merőben eltér a *Network File Systemtől (NFS)*, amelynél az ügyfélnek ismernie kell az *NFS* adott részének helyt adó fájlkiszolgálót. A különféle, független *AFS* tartományokat sejteknek nevezzük. A sejtek *Kerberos* tartományokhoz tartoznak. Egy jellegzetes *AFS* elérési út így néz ki: */afs/cern.ch/felhasznalo/a/alf/tervezetek/*. Az elérési útban szerepel ugyan az *AFS* sejt neve, ám a fájlkiszolgálóé nem.



1. ábra Az AFS fájltér mindenhonnan azonosnak látszik, használatakor az ügyfélnek nem kell tudnia, hogy melyik könyvtár melyik kiszolgálón található

A helyfüggetlenségnek köszönhetően az *AFS* rendszergazdák úgy mozgathatnak át fájlokat az egyik *AFS* kiszolgálóról a másikra, hogy a felhasználók ebből semmit nem észlelnek. Ugyancsak ez alapozza meg az *AFS* kiváló méretezhetőségét. Ha *AFS* fájlkiszolgálóinkon elfogy a hely, akkor elég egy újat üzembe állítanunk, majd áttelepítenünk rá az adatok egy részét. Az ügyfelek mindebből semmit nem vesznek észre. Az *AFS* a fájlkiszolgálónkénti felhasználók száma tekintetében is kiválóan méreteződik. Egy korszerű gépen egy *AFS* fájlkiszolgáló akár ezer ügyfelet is gond nélkül képes kiszolgálni.
A felhasználók szemszögéből az *AFS* fájltér pontosan úgy néz ki, mint az összes többi fájlrendszer. *Kerberos* hitelesítő adataikkal *AFS* alatt tárolt fájljaikat a világ bármely pontjáról elérhetik, köszönhetően az átfogóan egyedi névtérnek. Nézzünk egy példát: ha a svájci *CERN*-ben lévő kezdőkönyvtáramból a kaliforniai *SLAC*-ben lévő kezdőkönyvtáramba szeretnék másolni valamit, akkor két különböző *AFS*-sejt felé kell hitelesítenem magam:

```
% kinit --afslog alfw@ir.stanford.edu
alfw@ir.stanford.edu's Password:
% kinit -c /tmp/krb5cc_5828_1 --afslog alf@cern.ch
alf@cern.ch's Password:
```

Az *AFS* `tokens` parancsával meg tudjuk jeleníteni a hitelesítő adatokat:

```
% tokens Tokens held by the Cache Manager:
User's (AFS ID 388) tokens for afs@cern.ch [Expires
➥Apr  2 10:30]
User's (AFS ID 10214) tokens for
➥afs@ir.stanford.edu [Expires Apr  2 09:49]
--End of list--
```

A hitelesítés után mindkét kezdőkönyvtáramat el tudom érni:

```
% cp /afs/cern.ch/felhasznalo/a/alf/
➥tervezetek/X/src/hello.c \
   /afs/ir.stanford.edu/felhasznalok/a/l/alfw/
   ➥tervezetek/Y/src/.
```

Az *AFS* fájlkiszolgálók az adatokat különleges, */vicepXX* nevű lemezrészeken tárolják, ahol az XX az „a-zz" tartományba esik, vagyis egy-egy kiszolgálón összesen 256 lemezrész lehet. A lemezrészek mindegyikén adatkonténerek találhatók, ezeket köteteknek nevezzük. A kötetek a legkisebb mozgatásra, replikálásra vagy biztonsági mentés készítésére használható egységek. A köteteken belül könyvtárak és fájlok vannak. A kötetek csak azt követően válnak láthatókká, hogy befűzzük őket az *AFS* fájltérbe. Ezek a befűzési pontok pontosan olyanok, mint a könyvtárak.
Az *AFS* az ügyféloldali gyorsítótárazásnak köszönhetően különösen jól használható csak olvasható adatok, például a */usr/local/* fa tárolására. Az ilyen megoldások működésének javítását, üzembiztosságának fokozását az *AFS* azzal segíti, hogy lehetővé teszi a csak olvasható adatok másolatainak különböző *AFS* fájlkiszolgálókon való elhelyezését. Ha a példányokat tároló kiszolgálók valamelyike leáll, az ügyfél észrevétlenül átvált az adatok egy másik példányát tároló kiszolgálóra. Ugyanezzel a replikációs megoldással történik a földrajzilag távol lévő kiszolgálók közötti adatmásolás is. Az ügyfeleket be lehet úgy állítani, hogy lehetőleg a közeli másolatot használják, a távoli példányhoz csak meghibásodás esetén forduljanak. Az *openafs.org AFS* sejt például két kiszolgálóról fut, az egyik a *Pennsylvania* állambéli *Pittsburgh*-i *Carnegie Mellon Egyetemen*, a másik svédországi, stockholmi *Royal Institute of Technology (KTH)* intézetben található.
Az *AFS* pillanatképek készítésének lehetőségével segíti a biztonsági másolatok létrehozását. A pillanatképek létrehozása az általunk kiválasztott időpontban történik, és mindig kötetekről készülnek. A pillanatképeket ezután az egyéb felhasználói műveletek zavarása nélkül menthetjük le például szalagra, de külön befűzési pontokon, *AFS* kezdőkönyvtárukon belül akár a felhasználók számára is elérhetőkké tehetjük őket. Ezzel az egyszerű trükkel megelőzhetjük, hogy állandóan mentési és visszaállítási kérésekkel zaklassanak a felhasználók, hiszen az utolsó éjszaka készült pillanatképekben lévő fájlok szabadon hozzáférhetők.
Az *AFS* adatcsere protokollját nagy távolságú hálózatokhoz tervezték. Saját ***távoli eljáráshívás (remote procedure call, RPC)*** megvalósítást használ, ennek neve *Rx*, és *UDP* felett működik. A protokoll minden csomagkötetből csak az esetlegesen meghibásodott csomagokat küldi újra, és más protokollokhoz képest több nyugtázatlan csomag kint létét engedi meg.
Az *AFS* felügyelete bármelyik *AFS* ügyfélről elvégezhető, tehát semmi szükség nincs arra, hogy bármelyik *AFS* kiszolgálóra bejelentkezzünk. A rendszergazda tehát „szorosra zárhatja" az *AFS* kiszolgálókat, ami biztonsági szempontból kétségtelenül előnyös. Az *AFS* alatt tárolt adatok helyfüggetlensége szintén megkönnyíti a felügyelhetőséget; például egy *AFS* fájlkiszolgálót a köteteket más kiszolgálókra mozgatva akár teljesen – ráadásul a felhasználók számára észrevétlenül – ki tudunk üríteni. Az üres kiszolgálón ezután elvégezhetjük a szükséges karbantartásokat, legyen szó akár az operációs rendszer frissítéséről, akár a hardver javításáról. Amikor végeztünk, csak – ismét átlátszó módon – vissza kell mozgatnunk rá a köteteket.
Az *AFS* belül *Kerberost* alkalmaz a felhasználók hitelesítésére. Ez eredetileg a *Kerberos 4*-es változata, ám a Kerberos 5 bármelyik jelentősebb megvalósítása mellett is dönthetünk, ha fokozni szeretnénk a biztonságot. Az ***AFS hozzáférés-vezérlési listák (access control list, ACL)*** alapján korlátozza a könyvtárak elérését. Az *ACL*-ekben csak *Kerberos* főfiókok és ezek csoportjai szerepelhetnek, ellentétben az *NFS*-sel, amelynél kizárólag unixos felhasználóazonosítókkal történik a hitelesítés. Emellett egy további jogosultságkezelő szolgáltatás, a ***védelmi szolgáltatás (protection service, PTS)*** is figyelemmel követi az egyes *Kerberos* főfiókokat és főfiókcsoportokat.

Az AFS összetevői

Mindezen szolgáltatások biztosításához az *AFS*-nek számos különböző összetevőre van szüksége. Az *AFS* ügyfélprogramnak minden az *AFS* fájltér elérésére használni kívánt számítógépen futnia kell. Az *AFS* kiszolgáló program négy alaprészből áll. Hitelesítésre *Kerberost* használ, a jogosultságkezelést a *PTS* végzi, a kötethely kiszolgáló a helyfüggetlenséget teszi lehetővé, az adatszolgáltatásért pedig egy fájl- és egy kötetkiszolgáló felelős. Az egyes folyamatokat az *AFS* kiszolgálókon az ***alapszintű felügyelő (basic overseer, BOS)*** kiszolgáló kezeli. Mindezen feltétlenül szükséges összetevők mellett további démonok is léteznek az *AFS* kiszolgálók karbantartására és tartalmuk biztonsági mentésére. Az *AFS* kiszolgálók telepítése sajnos túlmutat témánkon. A különféle kiszolgáló összetevők miatt az *AFS* megismerésének első lépései egyenesen riasztók. Mégis érdemes megküzdeni vele, mára sok helyen létezni sem tudnának nélküle. Ha egy sejtet telepítettünk, akkor az *AFS* napi szintű karbantartását körülbelül 2 órában el tudjuk végezni, még nagyméretű telepítéseknél is.
Akit részletesebben is érdekel, hogy mások, például a *Morgan Stanley* és az *Intel* hogyan és mire használják az *AFS*-t, az vessen egy pillantást a legutóbbi *AFS Best Practices Workshopon* szerepelt bemutatóra (lásd az internetes forrásokat).

Az AFS ügyfelek telepítése

Az *AFS* kipróbálásához nincs szükség saját *AFS* kiszolgálóra. Elég, ha telepítjük az *OpenAFS* ügyfélprogramot, majd egy különleges, az idegen *AFS* sejtek nyilvánosan elérhető tereinek használatát engedélyező kapcsolóval indítjuk el az *AFS* ügyféldémont (`afsd`). Az *AFS* ügyfelek telepítésének legnehezebb mozzanata a szükséges rendszermagmodul beszerzése. Ha *Red Hat* vagy *Fedora* rendszert használunk, töltsük le a megfelelő *RPM*-eket (lásd a forrásokat). A rendszermagmodul mellett az *AFS* ügyfélnek egy felhasználói térbeli démonra (`afsd`), valamint az *AFS* parancskészletre is szüksége van. Ezek két további *RPM*-ben találhatók meg.
Ha megvannak a modulok, a következő lépés az *AFS*-ügyfél igényeink szerinti beállítása. Elsőként azt kell megadnunk,

© Kiskapu Kft. Minden jog fenntartva

© Kiskapu Kft. Minden jog fenntartva

hogy számítógépünk melyik sejt tagja legyen. Az *AFS* sejt neve a */usr/vice/etc/ThisCell* fájlban van megadva. Ha saját *AFS* kiszolgálókkal nem rendelkezünk, akkor a név tetszőleges, egyébként viszont az általuk kiszolgált sejt nevét kell beállítanunk. A következő beállítási érték az *AFS* helyi gyorsítótárával kapcsolatos. Az AFS-ügyfeleken az ügyfélprogramot külön lemezrészre kell telepíteni, a gyorsítótárat viszont bárhova tehetjük. A gyorsítótár helyét és méretét a */usr/vice/etc/cacheinfo* fájlban adhatjuk meg. A gyorsítótár alapértelmezett helye a */usr/vice/cache*, mérete pedig 100 MB, ami egy átlagos asztali vagy hordozható számítógép esetében bőségesen elegendő. Az *openafs-client RPM* telepítése után ezekkel a beállításokkal kezdünk, ilyenkor a *cacheinfo* fájlban az alábbi beállítást láthatjuk:

```
/afs:/usr/vice/cache:100000
```

Következőként az `afsd`, vagyis az *AFS* ügyféldémon a */etc/sysconfig/afs* fájlban található beállításait kell testreszabnunk. Az `OPTIONS` meghatározáshoz adjuk hozzá a `-dynroot` kapcsolót, így az AFS-ügyfél saját *AFS* kiszolgáló nélkül is el tud indulni.
Egy további fontos beállítás a `-fakestat`. Hatására az `afsd` meghamisítja a */afs/* könyvtárban lévő bejegyzések `stat(3)` adatait. Hiányában az *AFS* ügyfél világgá indulna, és minden számára ismert *AFS* sejttel kapcsolatba lépne. Ez jelenleg 133 sejtet jelent, amint a */afs/* könyvtárban kiadott hosszú listázással (`/bin/ls -l`) magunk is meggyőződhetünk róla.
Mivel az *AFS Kerberost* használ a hitelesítésre, gépünkön vagy gépeinken az időt is szinkronizálnunk kell. Az *AFS* ugyan saját szinkronizálási megoldással is rendelkezett, ám ez mára elavult, használatát nem javaslom. Kapcsoljuk is ki; ehhez a */etc/sysconfig/afs* fájl `OPTIONS` meghatározásához a `-nosettime` kapcsolót kell hozzáadnunk. Ha nincs kész megoldásunk az idő szinkronizálására, akkor ismerkedjünk meg a *hálózati idő protokollal (Network Time Protocol, NTP*; lásd a forrásokat).
Mindezen módosítások végrehajtása után a */etc/sysconfig/afs* fájl `OPTIONS` meghatározásának így kell kinéznie:

```
OPTIONS="$MEDIUM -dynroot -fakestat -nosettime"
```

Az utolsó lépés az *AFS* fájlrendszer befűzési pontjának létrehozása, amit a

```
% sudo mkdir /afs
```

parancs kiadásával végezhetünk el. Ezután a

```
% sudo /etc/init.d/afs start
```

paranccsal indíthatjuk el az AFS-ügyfelet. Az indítás eltart néhány másodpercig, ugyanis az `afsd`-nek indulása előtt fel kell töltenie a helyi gyorsítótárat. Mivel a gyorsítótár az újraindítások során is megőrzi tartalmát, a későbbi indítások már gyorsabbak lesznek.

Az AFS-világ felfedezése

Ha nincs is saját *AFS*-kiszolgálónk, de *AFS*-ügyfelünk beállításait a fentiek szerint módosítottuk, akkor az *AFS* használatával kapcsolatos parancsok egy részét, illetve a világszintű *AFS* teret egyedül is megismerhetjük.
Egy gyors ellenőrzéssel láthatjuk, hogy egyetlen *AFS* sejtben sem vagyunk hitelesítve:

```
% tokens
Tokens held by the Cache Manager:
--End of list--
```

A listában – a korábbi példával ellentétben – nincsenek hitelesítő adatok.
Első lépésként kérdezzük le a */afs/* könyvtár tartalmát. Ekkor az *AFS* ügyfelünk által ismert *AFS* sejtek mindegyike megjelenik. Most lépjünk át a */afs/openafs.org/software/openafs* könyvtárba, és listáztassuk ki ennek tartalmát is. A következőket kell látnunk:

```
% ls -l
total 10
drwxrwxrwx  3 root  root  2048 Jan 7  2003  delta
drwxr-xr-x  8 100   wheel 2048 Jun 23 2001  v1.0
drwxr-xr-x  4 100   wheel 2048 Jul 19 2001  v1.1
drwxrwxr-x 17 100   101   2048 Oct 24 12:36 v1.2
drwxrwxr-x  4 100   101   2048 Nov 26 21:49 v1.3
```

Lépjünk be valamelyik könyvtárba. Például:

```
% cd v1.2/1.2.10/binary/fedora-1.0
```

Vessünk egy pillantást a könyvtárban lévő *ACL*-ekre:

```
% fs listacl .
Access list for . is
Normal rights:
  openafs:gatekeepers rlidwka
  system:administrators rlidwka
  system:anyuser rl
```

A fentiekben két csoportot láthatunk, mindkettőhöz mind a hét jogosultság hozzá van rendelve: olvasás (`r`, *read*), keresés (`l`, *lookup*), beillesztés (`i`, *insert*), írás (`w`, *write*), teljes, önkéntes érvényű fájlzárolás (k) és ACL-módosítás (a).
A `system:anyuser` egy különleges, az AFS-hez tartozó csoport, olvasási (r) és keresési (l) joggal rendelkezik, így lényegében mindenkinek hozzáférést biztosít.
Egy csoport tagjait a `pts`, a védelmi szolgáltatás megfelelő parancsával adhatjuk meg:

```
% pts member openafs:gatekeepers -cell openafs.org
➥-noauth
Members of openafs:gatekeepers (id: -207) are:
  shadow
  rees
  zacheiss.admin
  jaltman
```

A `-noauth` kapcsolót azért használjuk, mert a parancsot erre a sejtre nézve hitelesítő adatok nélkül futtatjuk.
Az *AFS* hitelesítési részének – amely amúgy normál *Kerberos* – felderítéséhez különleges felügyeleti jogokra van szükség, ezért ettől most eltekintünk. Inkább nézzük meg, hogy a jelenlegi könyvtár fizikailag hol található:

```
% fs whereis .
File . is on hosts andrew.e.kth.se
➥VIRTUE.OPENAFS.ORG
```

A kimenet szerint a könyvtárból két példány létezik, az egyik az *andrew.e.kth.se*, a másik a *VIRTUE.OPENAFS.ORG* címen érhető el.

A
```
% fs lsmount /afs/openafs.org/software/openafs
?/v1.2/1.2.10/binary/fedora-1.0
/afs/openafs.org/software/openafs/v1.2/1.2.10/
➥binary/fedora-1.0
? is a mount point for volume #openafs.1210.f10
```

parancs kimenete szerint ez a könyvtár valójában az `openafs.1210.f10` nevű ***AFS*** kötet befűzési pontja.
A kötetet egy másik paranccsal vizsgálhatjuk meg:
```
% vos examine openafs.1210.f10 -cell openafs.org
➥-noauth
```

A fenti parancs az ***openafs.org AFS***-sejtben található `openafs.1210.f10` kötet írható-olvasható változatáról szolgáltat részletesebb adatokat. A kimenetnek így kell alakulnia:

```
openafs.1210.f10       536871770 RW   25680 K
➥On-line
    VIRTUE.OPENAFS.ORG /vicepb
    RWrite  536871770 ROnly  536871771 Backup     0
    MaxQuota          0 K
    Creation    Fri Nov 21 17:56:28 2003
    Last Update Fri Nov 21 18:05:30 2003
    0 accesses in the past day (i.e., vnode
    ➥references)

    RWrite: 536871770     ROnly: 536871771
    number of sites -> 3
       server VIRTUE.OPENAFS.ORG partition /vicepb
       ➥RW Site
       server VIRTUE.OPENAFS.ORG partition /vicepb
       ➥RO Site
       server andrew.e.kth.se partition /vicepb RO
       ➥Site
```

A kimenet értelmében a kötet a `VIRTUE.OPENAFS.ORG` állomás */**vicepb*** lemezrészén található. A következő sorban az írható-olvasható és a csak olvasható kötetek kötetazonosítói láthatók, némi statisztika társaságában. Az utolsó három sor a kötet egy írható-olvasható (`RW Site`) és két csak olvasható (`RO Site`) másolatának helyét adja meg.
Ha kíváncsiak vagyunk rá, vajon hány további ***AFS*** lemezrész található a `VIRTUE.OPENAFS.ORG` kiszolgálón, alkalmazzuk a következő parancsot:
```
% vos listpart VIRTUE.OPENAFS.ORG -noauth
```

Segítségével a következő lemezrészekről szerezhetünk tudomást:
```
/vicepa      /vicepb      /vicepc
Total: 3
```

Vagyis a gépen összesen három */**vicep*** lemezrész van.
Ha látni szeretnénk, hogy hány kötet található a kiszolgáló */**vicepa*** lemezrészén, adjuk ki az alábbi parancsot:
```
% vos listvol VIRTUE.OPENAFS.ORG /vicepa -noauth
```

A parancs végrehajtása eltart egy kis ideig, végül 275 kötet listáját látjuk. A lista első néhány eleme a következő:

```
Total number of volumes on server VIRTUE.OPENAFS.ORG
➥partition /vicepa: 275
openafs.10.src           536870975 RW   11407 K On-line
openafs.10.src.backup    536870977 BK   11407 K On-line
openafs.10.src.readonly  536870976 RO   11407 K On-line
openafs.101.src          536870972 RW   11442 K On-line
openafs.101.src.backup   536870974 BK   11442 K On-line
openafs.101.src.readonly 536870973 RO   11442 K On-line
```

Érdemes még ismerni a `bos` parancsot, amely egy sejt alapszintű felügyelő kiszolgálójával veszi fel a kapcsolatot, és meghatározza a rajta futó ***AFS*** kiszolgáló folyamatok állapotát. Az `fs`, a `pts`, a `vos` és a `bos` parancshoz számos további alparancs is tartozik. Szerencsére az AFS parancsainak mindegyike érti a `help` kapcsolót (elé nem kell kötőjelet írni), ennek segítségével megjeleníthetjük az alparancsokat is. Az `fs <alparancs> -help` (itt már kell a kötőjel) segítségével az egyes alparancsok szintaxisát is lekérdezhetjük.

Az AFS jövője

Jelenleg is számos az ***AFS*** fejlesztését célzó tervezet van folyamatban. A legfontosabb ezek közül az ***AFS***-nek a 2.6-os ***Linux*** rendszermagok alatti működését lehetővé tévő. Ezeknél a rendszermagoknál már nem érhető el szabadon a ***syscall*** tábla. Egy másik tervezet a kapcsolat nélküli üzemmód támogatását célozza, ennél az ***AFS***-ügyfelek hálózati kapcsolatuk megszakítása után is folytathatnák az ***AFS*** használatát, az ***AFS*** tér és a fájlok tartalma a kapcsolat ismételt létrejötte után kerülnének szinkronizálásra.

Összefoglalás

Bár az ***AFS***-t minden oldaláról egy csapásra megismerni gyakorlatilag lehetetlen, az ***AFS*** sejtek üzembe helyezésének folyamatát megismerni pedig komoly munka, az ***AFS***-t mégis kifizetődő saját infrastruktúránkon belül használni. Segítségével a biztonságos, gépfüggetlen, világszintű fájlmegosztás ugyanolyan könnyedén megoldható, mint a */**usr**/**local**/* könyvtárág vagy a unixos kezdőkönyvtárak tárolásának leegyszerűsítése. Ráadásul hosszú távon felügyeleti terheink növekedésével sem kell számolnunk.

Linux Journal 2005. április, 132. szám

A cikkhez tartozó források elérhetősége:

➲ ***www.linuxjournal.com/article/8079***

Dr. Alf Wachsmann 1999 óta a Stanford Linear Accelerator Center (SLAC) munkatársa. Ő felelős az önműködő Linux-telepítések minden mozzanatáért, egyaránt ide értve a farmok csomópontjainak, a kiszolgálóknak és az asztali gépeknek a kezelését. Munkája során elsősorban az aktív fájlkészletek (AFS) támogatásával, a Kerberos 5-re való áttéréssel, egy felhasználónyilvántartó tervezettel és felhasználói tanácsadással foglalkozik.

© Kiskapu Kft. Minden jog fenntartva