Fájlrendszer-felépítési Szabvány — Verzió 2.0

Fájlrendszer-felépítési Szabvány Csoport

szerkesztette: Daniel Quinlan

fordította: Murzsa Norbert

LaTeX: Szép Attila

Kivonat

Ez a szabvány a UNIX típusú operációs rendszerek számára megállapított, illetve előírt követelményeket és irányelveket tartalmazza a fájlok és a könyvtárak elhelyezésével kapcsolatban. Az irányelvek célja, hogy támogassák az alkalmazások belső kezelhetőségét, a szkripteket, a rendszeradminisztrációs, valamint a fejlesztési eszközöket éppúgy, mint a dokumentálás nagyobb egységességét ezen rendszerek számára.

1997. október 26.

1. Előszó

1.1. A szabvány jelenlegi állapota

A Fájlrendszer-felépítési Szabvány (*Filesystem Hierarchy Standard*, a továbbiakban FHS) verziószáma e dokumentumban 2.0 (FHS 2.0)

Az érdekeltek részéről szívesen fogadunk megjegyzéseket e szabványra nézve. A változtatásokra vonatkozó javaslatok formája tartalmazza az ajánlott változtatást a szövegre nézve, együttesen a megfelelő támogató megjegyzésekkel.

A szabvány valamennyi irányelve alárendelt az esetleges későbbi változtatásoknak. A dokumentumban található információkat mindenki csak a saját felelősségére használja.

1.2. A szabvány felépítése

Ez a szabvány a következő részekből áll:

1. Bevezető
2. A fájlrendszer: néhány fő alapelv megállapítása
3. A Root Könyvtár
4. A `/usr` felépítése
5. A `/var` felépítése
6. Operációs rendszer specifikus melléklet.

1.3. Megállapodások

Egy állandó szélességű font használt a fájlok és a könyvtárak nevének megjelenítésére.

Azon fájlnevek alkotóelemei, amelyek ábrázolásban eltérnek a tartalmi leírástól, kisebb-nagyobb zárójelekkel („`<`” és „`>`”) lesznek körülzárva, `<ezért>`. Az elektronikus levelek címei szintén kisebb-nagyobb zárójelek („<” és „>”) közé zártak, de megjelenítésük a dokumentum szokványos fontjával történik.

A fájlnevek választható alkotóelemei szögletes zárójelek („`[`” és „`]`”) között jelenítettek meg és kombinálhatóak a szokásos „`<`” és „`>`” jelekkel. Például, ha egy fájl létezhet kiterjesztéssel és anélkül is, akkor `<fájlnév>[.<kiterjesztés>]`-ként lesz megjelenítve.

A könyvtárak és a fájlok neveinek változtatható (változó) alrészei csillagjellel („`*`”) kijelzettek.

1.4. Az FHS háttere

A fájlrendszer-felépítés szabványának fejlesztési folyamata 1993 augusztusában kezdődött, azzal az erőfeszítéssel, hogy a Linux fájl- és könyvtár-struktúrája újraszerkesztett legyen. Az FSSTND (amelyet 1994. február 14-én adtak ki), egy olyan fájlrendszer-felépítési szabvány, amely speciálisan a Linux operációs rendszerre érvényes. A rákövetkező átdolgozott verziók 1994. október 9-én, majd 1995. március 28-án kerültek kiadásra.

1995 tavaszán a fejlesztés célja az volt, hogy az FSSTND egy még átfogóbb verzióját írják meg, amely nemcsak a Linux, hanem más UNIX alapú rendszerek által is elfogadott legyen; a BSD fejlesztői csoportjának tagjai segítségével. Eredményként egy összehangolt és öszpontosított kiadás készült el, amely általánosan érvényes volt a UNIX típusú rendszerekre. A kiszélesedő alkalmazási terület felismerése a szabvány nevének megváltozását is maga után vonta, amely így Fájlrendszer-felépítési Szabvány (*Filesystem Hierarchy Standard*) lett, röviden FHS.

1.5. Alkalmazási terület

Ez a dokumentum egy szabványos fájlrendszer-felépítést határoz meg az FHS fájlrendszerek számára, előírva a fájlok és a könyvtárak elhelyezkedését, valamint néhány fájl esetében a tartalmat is.

Ez a szabvány a rendszerintegrátorok, programcsomag-fejlesztők és rendszeradminisztrátorok számára készült, hogy felhasználják az FHS-hez alkalmazkodó fájlrendszerek felépítésében, valamint azok fenntartásában. A dokumentáció elsődleges törekvése a felvilágosítás és nem az oktatás, hogy hogyan kezeljenek egy hasonló típusú fájlrendszer felépítményt.

Az FHS kinőtte a korábbi szerepét, amelyet az FSSTND keretein belül foglalt el, azaz egy fájlrendszer szervezeti szabvány lett a Linux operációs rendszer részére. Beépült az FSSTND-be, címében hordozva a belső kezelhetőséget nem csak a Linux szervezet, hanem egy szélesebb kör (pl. a 4.4BSD alapú operációs rendszerek) számára is. Felöleli a tanulságokat a BSD világból, valamint további forrásokból is merített a multiarchitektúrák támogatásával és a heterogén hálózatok igényeivel kapcsolatosan.

Habár ez a szabvány még átfogóbb, mint az előző kísérlet a fájlrendszer felépítésének szabványosítására, periodikus frissítésére szükség lehet a felbukkanó technológiák változása miatt. Az is lehetséges, hogy az itt megnevezett problémákra hamarosan jobb megoldásokat találnak, vagy hogy a mi megoldásaink többé nem lesznek a legjobb megoldások, ezért e dokumentum periodikus frissítése közben előfordulhatnak majd utólagos javítások a továbbiakban is. Ezért aztán speciális cél a visszafelé kompatibilitás a dokumentum adott verziója és a következő között.

Örömmel fogadjuk a szabvánnyal kapcsolatos megjegyzéseket. Bármilyen megjegyzést vagy javaslatot a változtatással kapcsolatban közvetlenül az FHS szerkesztője (Daniel Quinlan <quinlan@pathname.com>) fogad, illetve küldhetők javaslatok még az FHS levelezőlistára is. A tipográfiával, illetve a helyesírással kapcsolatos megjegyzéseket közvetlenül az FHS szerkesztője fogadja.

Mielőtt azonban még kapcsolatba lépnél az FHS szerkesztőjével, előtte küldd el a leveledet a levelezési listára, hogy elkerüljük egy-egy esetleges régi téma túlzott újratárgyalását. (Természetesen a levelezőlista az oda nem illő üzeneteket nem fogja tárt karokkal fogadni.)

Esetenként felmerülhetnek kérdések a dokumentum egyes részeinek értelmezéseivel kapcsolatosan. Amennyiben ezek tisztázására szükséged van, kérjük, lépj kapcsolatba az FHS szerkesztőjével. Amióta ez a szabvány sok résztvevő egyezségét tükrözi, fontos megbizonyosodni arról, hogy valamennyi értelmezés mindnyájuk kollektív véleményét tükrözi. Emiatt nem lehetséges az azonnali válaszadás, hacsak a kérdés éppen nem az anyag előző vizsgálatakor lépett fel.

1.6. Általános irányelvek

Következzen néhány irányelv, amelyet szem előtt tartottunk e szabvány fejlesztésekor:

- megfejteni azokat a technikai problémákat, amelyek átmeneti nehézségként fellépnek,
- stabil leírás elkészítése,

- megszerezni a terjesztők, a fejlesztők, valamint más – a fejlesztői csoportokkal kapcsolatos – döntéshozók beleegyezését, és támogatni a részvételüket,
- létrehozni egy olyan szabványt, amely vonzó az eltérő UNIX típusú rendszerek alkalmazóinak, fejlesztőinek.

1.7. Kiknek szól e dokumentum?

E szabvány megcélzott közönségét a következő lista tartalmazza, bár természetesen nem korlátozódik kizárólagosan csak ezekre a csoportokra:

- rendszerfejlesztők
- rendszerintegrátorok és -terjesztők
- alkalmazásfejlesztők
- dokumentációírók
- rendszeradminisztrátorok és egyébb érdeklődő felek (információszerzés szempontjából)

1.8. Kivitelezés (implementáció)

A „kivitelezés" (*implementáció*) szó a terjesztésre (*disztribúció*), egy installált rendszerre, egy programra, egy csomagra, esetleg egy program vagy adat adott részére, komponensére utal.

Egy implementáció teljesen engedékeny a szabvány vonatkozásában, ha számára valamenyi feltétel együttesen található meg ebben a szabványban. Valamennyi fájl vagy könyvtár elhelyezkedésének – amely az implementáció részét képezi – meg kell egyeznie azokkal a helyekkel, melyek számukra a szabványban kijelöltek. Ha a szabványban egy fájl tartalma is meghatározott, akkor az implementációban található azonos fájl tartalma meg kell hogy feleljen ennek a leírásnak. Az implementációnak meg kell kísérelnie, hogy megtaláljon adott fájlokat vagy könyvtárakat (önmagán kívül), elsősorban persze azokon a helyeken keresve, amelyek meghatározottak e szabványban.

Egy implementáció teljesen kompatibilis ezzel a szabvánnyal, ha valamennyi általa tartalmazott fájl és könyvtár megtalálható azokon a helyeken, és ezek tartalma is megfelel azoknak az elvárásoknak, amelyek e dokumentumban meghatározottak. Még akkor is, ha a kérdéses fájloknak vagy könyvtáraknak nem az adott hely az elsődleges, illetve a fizikai elhelyezkedésük. Az implementációnak, amikor megkísérli megtalálni a számára szükséges fájlokat (amelyek önmagának nem részei), végig kell néznie azokat a helyeket (keresve a szükséges állományokat), amelyek e szabványban meghatározottak. Természetesen megkísérelhet állományokat keresni egyéb (nem e szabványban megadott) helyeken is.

Egy implementáció részlegesen engedékeny vagy kompatibilis ezzel a szabvánnyal, amennyiben teljesíti vagy kompatibilis e dokumentum jelentős részével. A részlegesen alkalmazkodó és a részlegesen kompatibilis kifejezéseket csak teljes terjesztések (*disztribúciók*) esetében alkalmazzuk és nem egy önálló program esetén. A „jelentős részével kompatibilis" kifejezés kétségkívül szubjektív vélemény, és – éppen emiatt – határesetekben az érdekelt félnek az FHS szerkesztőjével kellene kapcsolatba lépnie. Valószínűleg némely határeset elfogadott lesz.

Hogy alkalmassá tegyünk egy implementációt a „részlegesen FHS engedékeny" vagy a „részlegesen FHS kompatibilis" jelző kiérdemlésére szükséges ellátnunk azt egy mindenre kiterjedő listával, amely ismerteti az FHS dokumentummal szembeni különbségeit, majd ad egy rövid magyarázatot is arra, hogy miért volt szükség ezekre a változtatásokra. Ezt a listát mindig kapcsolni kellene az adott implementációhoz, valamint elérhetővé kellene tenni az FHS levelezőlista és az FHS szerkesztője számára is.

Az olyan kifejezéseket, mint a „szükséges”, „kellene”, „tartalmazza”, „legyen”, stb., az engedékenység és a kompatibilitás követelményeiként kellene értelmezni.

Mindamellett egy implementációnak nem szükséges valamennyi – e szabványban megadott – fájlt és könyvtárat tartalmaznia ahhoz, hogy engedékeny vagy kompatibilis legyen. Ez csupán azon fájlok számára szükséges, amelyeknek lelőhelyét e szabvány a megfelelő módon tartalmazza. Például, ha egy terjesztés (*disztribúció*) nem támogatja a minix fájlrendszert, akkor ennek segédprogramjait sem szükséges tartalmaznia, bár ezek világosan említésre kerülnek a `/sbin`-nel foglalkozó fejezetben.

Továbbá e dokumentum némely része tetszés szerint elhagyható. Természetesen ezek a részek világos módon meghatározottak vagy egy – esetleg több – „lehet”, „javasolt”, „tanácsolt” kifejezéssel jelzettek. Az opcionálisan kijelölt részek nincsenek hatással egy implementáció engedékenységére, illetve alkalmazkodására, hiszen ezek csupán javaslatokat jelentenek az általános gyakorlat támogatására és bárhol elhelyezhetők a végrehajtó (alkalmazó) választása alapján.

1.9. A fordításról

Nagyon sok leírás kezdődik valahogy így: „Alapművet tart kezében a mélyen tisztelt olvasó...” Nem szeretem az ilyen típusú mondatokat, mert ez nagyjából azt jelentené, hogy a különféle dokumentumok legalább 85–99%-a nélkül lehetetlen elboldogulni a nagyvilágban. Remélhetőleg én nem esem ebbe a hibába, miközben probálom elmagyarázni a mélyentiszteltnek, hogy miért is érdemes elolvasni ezt a dokumentumot.

Komolyra fordítva végre a szót...

A Linux Rendszeradminisztrátor Kézikönyv olvasása közben (köszönet Horváth András-nak <horvatha@kvark.szif.hu> a fordításért) többször találtam utalást egy FSSTND nevű valamire – mint forrás szabványra –, amely leírja a Linux operációs rendszer fájlrendszerét. Tudáséhségemtől vezérelve megkerestem e dokumentumot és nekifogtam az elolvasásának, amelyben igen sok – remélem nemcsak bennem felmerült – kérdésre kaptam választ.

Ráadásul a jószerencse látókörömbe sodort egy Szép Attila <szep@tolna.net> nevű úriembert, aki nélkül viszont biztosan nem készülhetett volna el a dokumentum magyar verziója, mivel LaTeX-ben jelen pillanatban 99%-os analfabéta vagyok, de már gyógyul... Köszönet neki a sok türelemért és a kitartásért, amivel a végefelé bennem is tartotta a lelket, hogy végre elkészüljön az FHS magyar verziója.

Szóval...

Alapművet tart a kezében a mélyen tisztelt olvasó... (na, pont erről beszéltem az előbb... :-(

Ezt a dokumentumot nem kötelező elolvasni. Aki nem szeretné tudni, hogy egy FHS (Fájlrendszer-felépítési Szabvány) kompatibilis vagy FHS engedékeny rendszerben melyik könyvtárban nagyjából milyen típusú programokat, fájlokat, könyvtárakat találhatunk, az most gyorsan hagyja abba az olvasást és lélegezzen fel, hogy sikerült elkerülnie egy – a számára – érdektelen iromány végigolvasását. Akik viszont mindig is szerették volna tudni, hogy pl. a Linux alatt mit hol találnak, vagy pontosabban mit, hol érdemes keresni, azok csemegére leltek. Hiszen ki az aki még soha sem veszett el `/var` könyvtárfa lombja között... „Mi a fene lehet a spool könyvtárban? És merre találom a nyomtatási várakozósorban lévő állományokat?”

Nos ezekre és a hasonló típusú kérdésekre a kedves olvasó itt megtalálhatja a választ. A leírtak természetesen nem 100%-ban érvényesek minden UNIX-típusú rendszerre (fordítás közben a DEBIAN és a RedHat Linux-ban is találtam eltéréseket ettől a dokumentumtól), hiszen az itt leírtak csak ajánlások és mindenki annyit fogad el belőle...

A legjobb az egészben az, hogy ezt a fájlrendszerleíró szabványt nem kell most rögtön elolvasni (azoknak különösen nem, akik már régen letették a kezükből) elejétől a végéig, mert bármikor előkereshető egy adott – mondjuk barangolás közben felfedezett – alkönyvtár funkciója, sőt esetleg a benne szabványosan megtalálható fájlok listája is. Így az emberfia akkor jut hozzá az információhoz, amikor az éppen fontos a számára és nem két szilvásgombóc közben a menzán.

Legalábbis. . .

(gondolom én :-)

Murzsa Norbert
E-mail: <kuksi@kuksi2.igyuk.hu>

2. A fájlrendszer

A UNIX fájlrendszer jellemzői:

- a hierarchikus felépítés
- a fájl adatainak következetes kezelése
- a fájl adatainak védelme

E szabvány feltételezi azt, hogy az operációs rendszer alapját egy olyan FHS-engedékeny fájlrendszer alkotja, amely támogatja azokat a biztonsággal kapcsolatos lehetőségeket, amelyek a legtöbb UNIX fájlrendszerben megtalálhatók. Ámbátor e szabvány nem kísérli meg, hogy összhangban legyen minden lehetséges szempontból bármely részleges UNIX rendszerű implementációval, azért – több szempontból is – a UNIX és a UNIX típusú rendszereken találtak alkotják a szabvány alapját alkotó elgondolásokat.

Ezek pedig (egyéb tényezők óvatos megfontolása után):

- a UNIX típusú rendszerek hagyományos és jól átgondolt gyakorlati alkalmazása
- más fájlrendszer-struktúrák kivitelezése
- alkalmazható szabványok

Definiálhatjuk a fájlok két világosan értelmezett kategóriáját: a megoszthatót a nem megoszthatóval szemben, és a változtathatót a nem változtatható tartalmú (*statikus*) ellenében.

A megosztható adatokat több hálózati számítógép megosztottan használhatja fel, míg a nem megosztható adatokat csak egy adott számítógép alkalmazhatja. Például a felhasználói home-könyvtárak lehetnek megosztottak, de az eszközmeghajtó lezárófájljai (*lock*) nem.

A statikus (nem változtatható tartalmú) fájlok közé tartoznak például a binárisok, a programkönyvtárak, a dokumentációk és minden más, amely a rendszergazda beavatkozása nélkül nem változik meg.

A mentés, az adminisztráció és a heterogén hálózati rendszerek közötti fájlmegosztás megkönnyítésére megkívánt, hogy egyszerű és könnyen megérthető legyen a feltérképezés a könyvtáraktól (különösen azon könyvtárak esetében, amelyek potenciális beillesztési – mountolási – pontok), egészen a bennük tárolt adatok típusáig.

E dokumentumon keresztül és természetesen bármely jól megtervezett rendszer esetében ezen legfőbb alapelvek megértése fogja segíteni a struktúra megértését és nyújt további következetességet.

A megosztható és a nem megosztható adatok közötti megkülönböztetés több okból is szükséges:

- Hálózati környezetben (például, ha több számítógép alkot egy szervezetet) igen nagy mennyiségű adat az, amely megosztott az eltérő gépek között (adatterületet spórolva ezáltal, valamint megkönnyítve a fenntartás folyamatát).
- Bizonyos fájlok hálózati környezetben is olyan adatokat tartalmaznak, amelyek csak egy adott számítógéphez kapcsolódnak. Ezért ezeket a fájlrendszereket nem lehet megosztani (mértéktelenül semmiképpen sem).
- A UNIX típusú fájlrendszerek történelmi implementációi egyazon fájlrendszerbe vegyítik a megosztható és a nem megosztható adatokat is, megnehezítve ezáltal a fájlrendszer széleskörű kiosztását.

A „megosztható" megkülönböztetés általában támogatott például:

- A `/usr` partíció (vagy annak egyes részei) beépítve – csak olvashatóan (*read-only*) – a hálózaton keresztül (használva az NFS-t /*Network File System*/).
- A `/usr` partíció (vagy annak egyes részei) beépítve csak olvasható adathordozóról. Egy CD-ROM például felhasználható egy csak olvasható fájlrendszer megosztására más FHS-engedékeny rendszerekkel, használva a postai levelezési rendszert hálózat gyanánt.

A „statikus"-sal (nem változó) szembeni „változtatható" megkülönböztetés két fő területen érinti a fájlrendszert:

- Amióta a `/` tartalmazza mindkét adattípust (a statikust és a változtathatót is), csak olvashatóként (*read-only*) szabad becsatlakoztatni.
- Miután a hagyományos `/usr` szintén tartalmazza mindkét adattípust (a statikust és a változtathatót is), és mivel csak olvashatóként (lásd feljebb) szeretnénk becsatlakoztatni, szükség van egy módszerre annak megoldására, hogy a `/usr` csak olvashatóként legyen beillesztve. Mindez megtehető a `/var` hierarchia létrehozásán keresztül, amely írhatóésolvashatóan csatlakoztatható be (vagy része lehet egy másik írhatóésolvasható partíciónak, pl. a `/`-nek), átvállalva a `/usr` partíció hagyományos szerepét.

Íme egy összefoglaló táblázat. Mivel e táblázat általános példákat tartalmaz, nem alkalmazható valamennyi lehetséges FHS-engedékeny rendszer implementációja esetében.

	megosztható	nem megosztható
statikus	/usr /opt	/etc /boot
változtatható	/var/mail /var/spool	/var/run /var/lock

3. A Root (gyökér) könyvtár

Ez a fejezet a root (gyökér) könyvtár felépítését írja le. A root fájlrendszernek – a megfelelő működéshez – tartalmaznia kell a bootolással, a mentésből való visszatöltéssel, az újraélesztéssel és a rendszer javításával kapcsolatos eszközöket:

- Egy rendszer bootolásakor, a root partíción már olyan eszközöknek kell lenniük, amelyekkel lehetséges becsatlakoztatni más fájlrendszereket is (például az együttműködni képes kiválasztott hardver és szoftver eszközök összessége, a bootolási információ, segédprogramok, egyéb, az induláshoz alapvetően szükséges adatok).
- A rendszer megjavításának és újraélesztésének lehetővé tételéhez olyan segédprogramok szükségeltetnek a root fájlrendszeren, amelyekkel egy tapasztalt rendszerfenntartó megvizsgálhatja és újraszerkesztheti a sérült rendszert.
- A rendszer visszatöltéséhez olyan segédprogramok szükségesek a root fájlrendszeren, amelyekkel elvégezhető egy – floppy-n vagy szalagos meghajtón stb. található – mentés visszatöltése.

A megfontolások mérlegelésekor az elsődleges gond az szokott lenni, hogy melyik a legkedvezőbb elhelyezése ennek a sok dolognak úgy, hogy továbbra is megtartsuk azt a célt, hogy a root fájlrendszer olyan kis méretű maradjon, amennyire csak lehetséges. Több okból is kívánatos, hogy a root fájlrendszer kisméretű legyen:

- A root fájlrendszert esetenként igen kis méretű adathordozón építjük fel.
- A root fájlrendszer igen sok rendszerspecifikus konfigurációs fájlt tartalmaz (például a rendszerre jellemző kernel, a számítógép neve, stb.). Ez azt jelenti, hogy a root fájlrendszer nem mindig osztható meg hálózati rendszerek között. Amennyiben sikerül kisméretűnek megtartani a hálózati rendszereken, ez minimalizálja a szervereken található, nem megosztható fájlokból adódó elveszett területek mennyiségét. Így lehetségessé válik, hogy a munkaállomások kisebb méretű helyi merevlemezzel rendelkezzenek.
- Ha a root fájlrendszert egy nagyméretű területen helyezed el, és teletöltöd mindenféle égi és földi jóval, akkor az ottani felhasználók kisebb területekkel fognak rendelkezni. Ha sok fájlt installáltál erre a nagyméretű root fájlrendszerre, akkor jóval gyakrabban is fogsz kompatibilitási hibákkal találkozni, ellentétben azokkal a rendszerekkel, amelyek a root fájlrendszert egy kisebb partíción helyezték el. Ha pedig fejlesztő vagy, akkor okosabb, ha az általad megszabott feltételek nem jelentenek probémát nagy számú felhasználó számára.
- A root fájlrendszeren érvénytelen adatokat előidéző lemezhibák jóval nagyobb problémát jelentenek, mint azok az adatsérülések, amelyek bármely más partíción lépnek fel. A kisméretű root fájlrendszer kevésbé hajlamos olyan sérülésre, amely esetleg még rendszerösszeomlást is eredményezhet.

A programoknak sosem kellene létrehozni vagy megkívánni olyan fájlokat vagy alkönyvtárakat, amelyek a root könyvtárban vannak. Ezek más elhelyezése az FHS hierarchiában jóval nagyobb hajlékonyságot adnak bármely programcsomag számára.

BEGIN ÉRTELMEZÉS

Különböző okai vannak annak, hogy miért tiltott új alkönyvtárak létrehozása a root fájlrendszeren:

- Plusz helyet foglal el azon a – root – partíción, amelyet a rendszeradminisztrátor minél egyszerűbbnek és kisebb méretűnek akar megtartani (például teljesítményi vagy biztonsági okból).

- Kívűl eshet a rendszeradminisztrátor figyelme alól, amikor az a szabvány fájlhierarchia osztályozását végzi a becsatlakoztatható köteteken keresztül.

END ÉRTELMEZÉS

/ — a root könyvtár

`bin`	alapvető parancs-binárisok
`boot`	a boot betöltő statikus fájljai
`dev`	eszközmeghajtó fájlok
`etc`	gépspecifikus rendszerkonfiguráció
`home`	felhasználói home könyvtárak
`lib`	alapvető megosztott programkönyvtárak és kernel modulok
`mnt`	az ideiglenesen csatlakoztatott partíciók beillesztési pontja
`opt`	további alkalmazási programcsomagok
`root`	a root felhasználó home könyvtára
`sbin`	alapvető rendszer-binárisok
`tmp`	ideiglenes fájlok
`usr`	másodlagos hierarchia
`var`	változó adatok

A fenti listában meghatározott valamennyi könyvtár részletezésre kerül az alábbi alfejezetekben. A `/usr` és a `/var` könyvtárak felépítését egy-egy külön fejezet ismerteti.

Az operációs rendszer kernel-image fájlját vagy a `/` vagy a `/boot` könyvtárban kell elhelyezni. A kernel elhelyezkedésének további információi a `/boot` könyvtárra vonatkozó részben találhatók.

3.1. /bin : Alapvető felhasználói parancs binárisok (valamennyi felhasználó részére)

A `/bin` könyvtár azokat a parancsokat tartalmazza, amelyek a rendszeradminisztrátor és a felhasználók által egyaránt használtak, de amelyek szükségesek az egyszemélyes (*single*) felhasználói módban is. Olyan parancsokat is tartalmazhat, amelyek közvetett módon – scriptek által – használhatók.

A `/bin` könyvtárnak ne legyen alkönyvtára.

Azokat a parancs-binárisokat, amelyek viszonylag nélkülözhetőek, inkább a `/usr/bin` könyvtárban kellene elhelyezni a `/bin` helyett. Azok az adatok, amelyek inkább a normál felhasználók számára szükségesek (`mail`, `chsh`, stb.), általában nélkülözhetőek annyira, hogy ne a root partícióra kerüljenek.

A /bin legszükségesebb fájljai:

- általános parancsok:

  A következő parancsok mindenképpen szükségesek a `/bin` számára. Most csak néhányat sorolunk fel azon parancsok közül, amelyeknek helye hagyományosan a `/bin` könyvtárban van.

`{ cat, chgrp, chmod, chown, cp, date, dd, df, dmesg, echo, ed, false, kill, ln, login, ls, mkdir, mknod, more, mount, mv, ps, pwd, rm, rmdir, sed, setserial, sh, stty, su, sync, true, umount, uname }`

Amennyiben a `/bin/sh` a Bash-t jelöli, akkor mutasson egy merev vagy szimbolikus link a `/bin/bash`-ra, mivel a Bash különbözőképpen viselkedik, ha az `sh`-val hívjuk meg, mint amikor a `bash`-val. A `pdksh` szintén lehet a `/bin/sh` az installációs lemezen, ezért ezt is érdemes úgy rendezni, hogy a `/bin/sh` egy szimbolikus link legyen a `/bin/ksh`-ra. Az ilyen esetekben a szimbolikus link használatával a felhasználó könnyen láthatja azt, hogy a `/bin/sh` nem a valódi Bourne parancsértelmező.

Amióta a C-parancsértelmező ténylegesen szabványos helye a `/bin/csh`, (ha – és csak, ha a C- vagy egy vele megegyező (például a `tcsh`) parancsértelmező – elérhető a rendszeren), ezt mindenképpen a `/bin/csh` néven kell elérhetővé tenni. A `/bin/csh` lehet egy szimbolikus link a `/bin/tcsh`-ra, vagy a `/usr/bin/tcsh`-ra.

Megjegyzés: A `[` *és a* `test` *parancsok beépítettek a legtöbb általánosan használt Bourne parancsértelmező (*`/bin/sh`*) alkotóelemei közé. Ezt a két parancsot tehát nem kell a* `/bin`*-be helyezni; ezek a* `/usr/bin`*-be kerülnek. Ez a két parancs azért létezik különálló formában is, hogy megkíséreljen eleget tenni a POSIX.2 szabvánnyal rendelkező UNIX vagy UNIX-típusú rendszereknek is.*

- a visszaállítást szolgáló parancsok:

  Ezek a parancsok lehetővé teszik egy rendszer visszaállításának lehetőségét (ellátva ezzel a `/` sértetlenségét).

  `{ tar, gzip, gunzip` (link a gzip-re)`, zcat` (link a gzip-re) `}`

  Ha a rendszer mentések végrehajtására a `gzip`-től és a `tar`-tól eltérő program használatos, akkor a root partíciónak tartalmaznia kell a visszaállításhoz szükséges minimális összetevőket. Például sok rendszer tartalmazza a `cpio` parancsot, amely a másik legismertebben használt mentési segédeszköz a `tar` után. Természetesen, ha nem szükséges a root partíció mentése, ezek a bináris állományok elhagyhatók (például, ha egy root ROM chip illeszti be a `/usr` fájlrendszert az NFS-en keresztül). Amennyiben a rendszer visszaállítását NFS-en keresztül kívánjuk megvalósítani, akkor az `ftp` vagy a `tftp` programot kellene elérhetővé tenni (az ftp kapcsolathoz szükséges valamennyi összetevővel együtt) a root partíción.

  A különböző rendszereken a visszaállításhoz szükséges parancsok kerülhetnek a `/bin` vagy a `/usr/bin` könyvtárba.

- hálózati parancsok:

  Itt csak azok a binárisok találhatók, amelyek a felhasználók és a root számára is egyformán szükségesek lehetnek. További hálózati parancsok vannak még a `/usr/bin`, esetleg a `/usr/local/bin` könyvtárakban.

  `{ domainname, hostname, netstat, ping }`

3.2. /boot : A boot betöltő statikus (nem változtatható) fájljai

Ez a könyvtár tartalmazza mindazt, ami szükséges a bootolás folyamatához, kivéve a konfigurációs fájlokat és a térképinstallálót (*map installer*). Ezért a `/boot` azokat a fájlokat tárolja, amelyek akkor kerülnek használatra, mielőtt még a kernel elkezdené futtatni a felhasználói-módú parancsokat. Ez a könyvtár tartalmazhatja a lementett master boot rekordot, a szektor térkép fájlokat és olyan további adatokat, amelyek nem szerkeszthetők közvetlenül kézzel. Azokat a programokat, amelyek ahhoz szükségesek, hogy a boot-töltő képes legyen betölteni egy fájlt, a `/sbin` könyvtárban kell elhelyezni. A boot-töltők konfigurációs fájljainak helye a `/etc` könyvtárban van.

Az operációs rendszer kerneljét a / vagy a /boot könyvtárban kell elhelyezni.

Megjegyzés: Némely i386-os számítógépen szükséges lehet, hogy a /boot *egy különálló partíción legyen elhelyezve, elkerülve így a boot-eszközmeghajtó 1024 cilinderszámánál nagyobb értéket, amely a hardverkorlátozásoknak köszönhető.*

3.3. /dev : Eszközmeghajtó fájlok

A /dev könyvtár speciális vagy eszközmeghajtó fájlok lelőhelye.

Amennyiben lehetséges, a /dev könyvtár eszközmeghajtó fájljait kézi úton kell létrehozni. A /dev könytárnak tartalmaznia kell egy MAKEDEV nevű parancsot, amellyel a szükséges eszközmeghajtó fájlok létrehozhatók. A könyvtár tartalmazhat még egy MAKEDEV.local nevű fájlt is a helyi eszközmeghajtó fájlok számára.

Ha megkívánt, akkor a MAKEDEV segítségével kell létrehozni azokat az eszközmeghajtó fájlokat, amelyek a rendszeren találhatók és nem csak azokat, amelyeket egy különálló implementáció hoz létre.

3.4. /etc : Gépspecifikus (egyedileg jellemző) rendszerkonfiguráció

A /etc könyvtár olyan konfigurációs fájlokat és könyvtárakat tartalmaz, amelyek egyedileg jellemzőek egy adott rendszerre.

A /etc könyvtárban nem szükséges bináris fájlokat elhelyezni.

/**etc** — Gépspecifikus rendszerkonfiguráció (beállítások)

```
|
├── X11        az X Window rendszerre vonatkozó beállítások
└── opt        a /opt könyvtárra vonatkozó beállítások
```

A következő bekezdés szándéka – néhány példán keresztül – részben megvilágítani a /etc könyvtár tartalmának leírását (a lista persze nem feltétlenül teljes).

A /etc könyvtár számára megkívánt fájlok:

- általános fájlok:

  { adjtime, csh.login, disktab, fdprm, fstab, gettydefs, group, inittab, confissue, ld.so.conf, lilo.conf, motd, mtab, mtools, passwd, profile, securetty, shells, syslog.conf, ttytype }

- hálózati fájlok:

  { exports, ftpusers, gateways, host.conf, hosts, hosts.allow, hosts.deny, hosts.equiv, hosts.lpd, inetd.conf, networks, printcap, protocols, resolv.conf, rpc, services }

Megjegyzés:

A bootolási idő alatt használt parancsszkriptek beállítása hasonló a System V vagy a BSD modelljeihez. Ennek a területnek további leírása kapcsolt lesz e szabvány jövőbeni verzióihoz.

*A jelszó árnyékolási készletet (*shadow password*) használó rendszereknek további fájljai lehetnek a* /etc *könyvtárban (*/etc/shadows *és egyéb állományok) és programjai a* /usr/sbin *könyvtárban (*useradd, usermod *és egyéb állományok).*

3.4.1. /etc/X11 : Az X Window rendszerre vonatkozó beállítások

A `/etc/X11` a javasolt elhelyezés valamennyi X11 gépspecifikus konfiguráció számára. Ezt a könyvtárat helyi ellenőrzés alatt szükséges tartani, amennyiben a `/usr` csak olvasható (*read only*) módon illesztett be. Az ebben a könyvtárban szükséges fájl az `Xconfig` (és/vagy `XF86Config`), valamint az `Xmodmap`.

A `/etc/X11` alkönyvárjai között vannak az `xdm` és egyéb más programok (például az ablakkezelő programok) könyvtárai. Javasoljuk, hogy az ablakkezelő programoknak csak egy-egy konfigurációs fájlja legyen itt, amelyek alapértelmezett `.*wmrc` neve inkább `system.wmrc` legyen (hacsak a neve nem egy széles körben elfogadott vagylagos név) és ne használjon külön alkönyvtárat. Bármely ablakkezelő alkönyvtárainak neve meg kellene, hogy egyezzen az aktuális ablakkezelő binárisával.

`/etc/X11/xdm` tárolja a konfigurációs fájlokat az `xdm` számára. Ezen fájlok legtöbbje normális esetben megtalálható a `/usr/lib/X11/xdm` könyvtárban. Néhány helyi, az `xdm`-mel kapcsolatos változó adat a `/var/state/xdm` könyvtárban tárolt.

3.4.2. /etc/opt : A /opt könyvtár konfigurációs állományai

A hozzáadott alkalmazási programcsomagok gépspecifikus konfigurációs fájljait a `/etc/opt/<csomag>` könyvtárban kellene elhelyezni, ahol a `<csomag>` annak a részfának a kezdete a `/opt` könyvtáron belül, ahol a csomag statikus adatai tároltak. Nincs külön előírt struktúra a `/etc/opt/<csomag>` belső elrendezésére.

Ha a konfigurációs fájlt ettől eltérő könyvtárban szükséges elhelyezni a csomag működése szempontjából, vagy hogy a rendszer ténylegesen működjön, elhelyezhető a `/etc/opt/<csomag>`-tól eltérő helyen is.

BEGIN ÉRTELMEZÉS

További információért fordulj a `/opt` értelmezéséhez.

END ÉRTELMEZÉS

3.5. /home : Felhasználói home könyvtárak (szabadon választható)

A `/home` viszonylag szabvány fogalom, de valójában ez egy adott számítógépre jellemző fájlrendszer, amely beállítás gépről gépre eltérő. Ez a rész csak egy javasolt elhelyezést ír le a felhasználói home könyvtárak számára, mindazonáltal tanácsoljuk, hogy valamennyi FHS-engedékeny terjesztés (*disztribúció*) ezt használja a home könyvtárak alapértelmezett elhelyezéseként.

Kisebb rendszereken minden felhasználó könyvtára tipikusan csak egy a `/home` könyvtár nagyszámú alkönyvtára között. (Például: `/home/smith`, `/home/torvalds`, `/home/operator`, stb.)

Nagyobb rendszereken (főképp amikor a home könyvtárak NFS-en keresztül nagy számú számítógép között megosztottak) hasznos lehet alosztályokba sorolni a felhasználói home könyvtárakat. Az alosztályokba sorolás megvalósítható például olyan könyvtárneveket használva, mint `/home/staff`, `/home/guests`, `/home/students`, stb.

Különböző emberek igen eltérő helyeket részesítenek előnyben a felhasználói számlák (*account*) elhelyezésénél. Ezért javasolt, hogy a programok ne építsenek erre a fellelési helyre. A felhasználói home könyvtár kitalálásához inkább használjuk a `getpwent(3)` programkönyvtár funkcióját, mintsem a `/etc/passwd` fájl információit, mivel a felhasználói információk tárolhatóak távoli módon is, használva olyan rendszereket, mint például a NIS.

3.6. /lib : Alapvető megosztott programkönyvtárak és kernel modulok

A `/lib` könyvtár azokat a megosztott programkönyvtár képfájlokat (*shared library images*) tartalmazza, amelyek a rendszer bootolásához szükségesek és a programokat futtatják a root fájlrendszeren.

/**lib** — Alapvető megosztott programkönyvtárak és kernel modulok

`modulok`	Betölthető kernel modulok

Itt található a `/lib/libc.so.*`, a `/lib/libm.so.*`, a megosztott dinamikus kapcsoló (*shared dynamic linker*) `/lib/ld.so`, és további – a `/bin` és a `/sbin` könyvtárak bináris állományai számára szükséges – megosztott programkönyvtárak.

Azok a megosztott programkönyvtárak, amelyek a `/usr` könyvtár bináris állományai számára szükségesek (például bármely X Window bináris), nem tartoznak a `/lib` könyvtár alá. Kizárólag a `/bin` és a `/sbin` könyvtárak bináris állományainak futásához szükséges megosztott programkönyvtárakat kellene ebben a könyvtárban elhelyezni. A `libm.so.*` programkönyvtár elhelyezhető még a `/usr/lib` könyvtárban is, ha nem szükséges sem a `/bin`, sem a `/sbin` könyvtárak fájljai számára.

Kompatibilitási okok miatt szükséges egy hivatkozás megléte a rendszeren található C-előfeldolgozóra (*C preprocessor*), `/lib/cpp` néven. E bináris szokványos megtalálási helye a `/usr/lib/gcc-lib/`
`<a`$_{m}egclzott_{a}rchitektra >$ / $< verzi >$
$/cpp.A/$`lib/cpp`$vagyerreabinrisramutat, vagyegytetszlegesnevhivatkozsonkeresztlrhetelarendszerenmeglvbinrisllomny($*pldula*`/usr/lib/cpp`*isigengyakranhasznlt*).

A `/lib/modules`-vel kapcsolatos előírások előkészületben.

3.7. /mnt : Beillesztési pont az időlegesen becsatlakoztatott fájlrendszerek részére

E könyvtár funkciója, hogy a rendszeradminisztrátor számára időlegesen szükséges fájlrendszerek becsatlakoztatási helyét ellássa. A könyvtár tartalma lokális kérdés, ezért ezt nem kellene módosítani, illetve nem kellene építeni egy esetleges tartalom meglétére.

Javasoljuk, hogy az installációs programok ne használják ezt a könyvtárat és hogy ne használjunk egy alkalmasnak tűnő ideiglenes könytárat a rendszer által használt helyett.

3.8. /opt : Hozzáadott alkalmazási programcsomagok

/**opt** — Hozzáadott alkalmazási programcsomagok

`<csomag>`	Statikus csomag objektumok

A `/opt` a hozzáadott alkalmazási programcsomagok részére fenntartott.

A csomagot a `/opt` könyvtárba installáljuk és a statikus fájljait egy különálló `/opt/<csomag>` könyvtárfában helyezzük el (ahol a `<csomag>` az a név, amely leírja a szoftvercsomagot).

Azokat a programokat, amelyek a felhasználók által használtak a `/opt/<csomag>/bin` könyvtárban kellene elhelyezni. Amennyiben a csomag tartalmaz UNIX kézikönyvlapokat (*manual pages*), ezek a

`/opt/<csomag>/man` könyvtárba tehetők, hasonló strukturában, mint ami a `/usr/share/man` könyvtárban használt.

A `/opt/bin`, a `/opt/doc`, a `/opt/include`, a `/opt/info`, a `/opt/lib` és a `/opt/man` könyvtárak fenntartottak a helyi rendszeradminisztrátor használatára. A csomagok által létrehozott előtér fájlok (*front-end files*) ezekbe a – rendszeradminisztrátor számára fenntartott – könyvtárakba legyenek elhelyezve, akár hivatkozás (*link*), akár másolat (*copying*) formájában, de a normális működésében ne legyen változás ezen fenntartott könyvtárak hiányában sem.

A csomagok változó tartalmú fájljai – amelyek a normál működés alatt változnak – a `/var/opt` könyvtárba legyenek installálva. További információért fordulj a `/var/opt` könyvtárral foglalkozó részhez.

A gépspecifikus (egyedileg jellemző) konfigurációs fájlokat a `/etc/opt` könyvtárba kellene installálni. További információk ezzel kapcsolatban a `/etc` könyvtárt tárgyaló fejezetben.

Ne helyezzünk el a `/opt`, a `/var/opt` és a `/etc/opt` hierarchiákon kívül további – a csomaghoz tartozó – fájlokat, kivételek ez alól azok a csomagfájlok, amelyeket a pontos működés megvalósítása érdekében szükséges a fájlrendszerfa egy speciális részében elhelyezni. Például az eszközmeghajtók lezárófájljait (*device lock files*) a `/var/lock` könyvtárban szükséges elhelyezni, az eszközmeghajtó fájlokat pedig a `/dev` könyvtárban.

BEGIN ÉRTELMEZÉS

A hozzáadott alkalmazási programcsomagok részére fenntartott `/opt` könyvtár használata jól megalapozott gyakorlattal rendelkezik a UNIX közösségben. A System V Alkalmazási Bináris Csatolóegysége (System V Application Binary Interface) [AT&T 1990] , amely a System V Csatolóegység Meghatározás harmadik kiadásán alapul (System V Interface Definition – Third Edition), hasonló struktúrát ad meg az imént ismertetetthez.

Az Intel Bináris Kompatibilitási Szabványa v.2 (The Intel Binary Compatibility Standard vagy iBCS2) szintén hasonló struktúrát határoz meg a `/opt` részére.

Általában a `/opt/<csomag>` könyvtáron belül kellene elhelyezni mindazokat az adatokat, amelyek szükségesek a rendszeren található adott csomag támogatásához, beleértve ebbe azokat a fájlokat is, amelyek másolatai a `/etc/opt/<csomag>` és a `/var/opt/<csomag>` könyvtárakba kerülnek, csakúgy mint a fenntartott alkönyvtárakat a `/opt` fában.

END ÉRTELMEZÉS

3.9. /root : A root felhasználó home könyvtára (elhagyható)

A root számlához (*account*) tartozó hagyományos home könyvtár a UNIX rendszereken a `/`. A `/root` könyvtárat igen sok Linux és néhány UNIX rendszeren is használják (például csökkentve ezzel a `/` könyvtárban található zűrzavart). A root home könyvtára a fejlesztő vagy a helyi előnyök által is meghatározott lehet. Kézenfekvő választás lehet például a `/`, a `/root` vagy a `/home/root` könyvtár.

Ha a root home könyvtára nem a root partíción található, ezt mindenképpen szükséges létrehozni alapértelmezésben a `/` könyvtárat megadva, amennyiben ez nem megadott.

Megjegyzés: Ne használjuk a root számlát olyan evilági dolgokra, mint például a mail vagy a news, hanem kizárólag csak a rendszer adminisztrálására, karbantartására vegyük igénybe. Emiatt ne is legyenek olyan alkönyvtárak a root home könyvtárában, mint a `Mail` *vagy a* `News`*, és azok a levelek, amelyek az adminisztráció számára a root-hoz, postmaster-hez, webmaster-hez címzettek, inkább legyenek átirányítva (*forward*) egy megfelelő felhasználóhoz.*

3.10. /sbin : Rendszerbinárisok, (amelyeknek egykor a /etc könyvtárban volt a helyük)

A rendszeradminisztrációra használt segédprogramok (és egyéb – csak a root által használt – programok) a `/sbin`, `/usr/sbin` és a `/usr/local/sbin` könyvtárakban tároltak. A `/sbin` könyvtár tipikusan csak a rendszer bootolásához és a `/bin` könyvtár binárisaihoz szükséges alapvető programokat tartalmazza. Bármilyen program is kerül futtatásra a `/usr` beillesztése után (amennyiben nem lép fel semmilyen proléma), azokat már a `/usr/sbin` könyvtárba kellene elhelyezni. A `/usr/local/sbin` könyvtárba pedig kizárólag a helyi rendszer adminisztrálására fenntartott bináris állományokat kellene elhelyezni.

Rendkívül egyszerű annak a kérdésnek az eldöntése, hogy mely dolgok is kerüljenek a `/sbin` könyvtárba: ha azt egy hétköznapi (nem a rendszeradminisztrátor) felhasználó a későbbiekben bármikor is közvetlenül használni fogja, akkor tegyük az egyik „`bin`" könyvtárba. A hétköznapi felhasználók elérési útjába (*path*) nem kellene egyik `sbin` könyvtárat sem betenni.

Megjegyzés: Az olyan fájlokat, amelyeket a felhasználók viszonylag ritkán használnak (például a `chfn`*), szintén még a* `/usr/bin` *könyvtárba kellene elhelyezni. Bár a* `ping` *abszolút szükséges a root számára (például a hálózat helyreállítása és a diagnosztika közben), de ezt a parancsot gyakran használják a normál felhasználók is, emiatt a* `/bin` *könyvtárban kellene lennie.*

Javasoljuk, hogy a felhasználóknak legyen olvasási és futtatási joguk valamennyi állományra a `/sbin` könyvtárban, kivétel ez alól talán néhány setuid-s és setgid-s program. A megosztás a `/bin` és a `/sbin` könyvtár között nem biztonsági okokra vezethető vissza, sőt nem is amiatt történt, hogy akadályozza az operációs rendszer átlátását a felhasználók számára. Inkább azért, hogy ellásson egy felosztást azok között a binárisok között, amelyeket mindenki használ és azok között, amelyek elsősorban adminisztrációs folyamatokra használatosak. Semmiféle biztonsági előny nem származik abból, ha a `/sbin` könyvtár korlátozva van a felhasználók részére.

A /sbin könyvtár számára megkívánt fájlok:

- általános parancsok:

  `{ clock, getty, init, update, mkswap, swapon, swapoff, telinit }`

- a rendszer leállításával kapcsolatos parancsok:

  `{ fastboot, fasthalt, halt, reboot, shutdown }`

  (Vagy a fenti felsorolás bármilyen kombinációja egészen addig, amíg a `shutdown` is beleértett.)

- parancsok a fájlrendszer karbantartására:

  `{ fdisk, fsck, fsck.*, mkfs, mkfs.* }`

  A `*` jel egyenlő lehet az `ext`, `ext2`, `minix`, `msdos`, `xia` értékek valamelyikével, illetve egyéb értékeket is felvehet.

- hálózati parancsok:

  `{ ifconfig, route }`

3.11. /tmp : Ideiglenes fájlok

A `/tmp` könyvtárat azon programok számára kellene elérhetővé tenni, amelyeknek működéséhez szükségesek ideiglenes (*temporary*) fájlok.

Habár a `/tmp` könyvtárban tárolt adatok gépspecifikus (egyedien jellemző) módon törlődnek, ám mindenképpen javasolt, hogy az itt tárolt könyvtárak és fájlok a rendszer bootolása alatt kerüljenek törlésre.

A programok ne feltételezzék azt, hogy a `/tmp` könyvtárban megőrzésre kerül bármiféle fájl vagy könyvtár, egy adott program eltérő idejű használatai között.

BEGIN ÉRTELMEZÉS

Az IEEE szabvány P1003.2 (POSIX, 2.rész) a fentiekhez hasonló követelményeket fogalmaz meg.

Az FHS által hozzáadott javaslat, azaz hogy a `/tmp` könyvtár tartalma a bootolás ideje alatt kerüljön törlésre, történelmi alapokon és a mindennapos gyakorlaton nyugszik, ám ez nem követelmény, mivel a rendszeradminisztráció nem tartozik bele e szabvány alkalmazási területébe.

END ÉRTELMEZÉS

4. A /usr hierarchia

A `/usr` a fájlrendszer második legfontosabb része. A `/usr` megoszható, csak olvasható (*read-only*) adatokból áll. Ez azt jelenti, hogy a `/usr` hierarchiának megosztottnak kellene lennie azok között a különböző számítógépek között, amelyek FHS-engedékenyként futnak, valamint nem kellene írhatónak lennie. Bármilyen információt, amely gépspecifikus vagy esetleg idővel változik, ettől eltérő helyen kellene tárolni.

Nem sok szoftvercsomagnak kellene használnia a `/usr` hierachia alatt található közvetlen alkönyvtárakat. Kivételt képez ez alól az X Window Rendszer (*the X Window System*), mivel tekintélyes kivételt képez és mert gyakorlata széles körben elfogadott. A szabvány ezen szakasza részletezi a legtöbb dolog (például a csomagok) elhelyezkedését a hierarchián belül.

/**usr** — a másodlagos felépítmény (*hierarchy*)

`X11R6`	az X Window rendszer 11. verziója és 6. kiadása (ver. 11, rel. 6)
`X386`	az X Window rendszer 11. verziója és 5. kiadása az x86 platformokon
`bin`	a legtöbb felhasználói parancs
`games`	játékok és oktatással kapcsolatos binárisok
`include`	fejléc fájlok a C programok részére
`lib`	programkönyvtárak
`local`	helyi hierarchia (a fő installálás után üres)
`sbin`	nem létfontosságú rendszerbinárisok
`share`	architektúra-független adatállományok
`src`	forráskód

A következő könyvtárakra mutató, szimbolikus hivatkozásoknak szükséges a megléte. Ezek azon a kívánalmon alapulnak, hogy szükséges megőrizni a kompatibilitást a régebbi rendszerekkel egészen addig, amíg valamennyi implementáció feltételezetten a `/var` hierarchiát használja.

`/usr/spool -> /var/spool`

`/usr/tmp -> /var/tmp`

`/usr/spool/locks -> /var/lock` A későbbiekben, ha már egy szimbolikus hivatkozás (*link*) sem szükséges a fentiek közül, a hivatkozások törlésre kerülhetnek, ha ez megkívánt.

4.1. /usr/X11R6: az X Window Rendszer, 11. verzió, 6. kiadása

Ez a hierarchia a X Window Rendszer, 11. verzió, 6. kiadása és az ehhez kapcsolódó fájlok részére van fenntartva.

A dolgok egyszerűsítése végett és hogy az XFree86-ot kompatibilisabbá tegyük a más rendszereken lévő X Window Rendszerrel, szükséges a következő szimbolikus hivatkozások (*links*) megléte:

`/usr/bin/X11 -> /usr/X11R6/bin`

`/usr/lib/X11 -> /usr/X11R6/lib/X11`

`/usr/include/X11 -> /usr/X11R6/include/X11`

Általában szoftvert nem kellene installálni vagy karbantartani a fenti szimbolikus kötéseken (*links*) keresztül. Ezek csak felhasználói hasznosításra szántak. A nehézség az X Window Rendszer kiadott verziójával kapcsolatban – ami csak átmeneti időközönként lép fel –, hogy nem lehet azt tudni, hogy az X11 melyik verziója van használatban.

A /usr/X11R6/lib/X11 könyvtárban található gépspecifikus adatokat csak demonstrációs fájlokként kellene értelmezni, mivel az alkalmazások által az adott számítógépről megkövetelt információknak (olyan fájlokból, mint az Xconfig, XF86Config vagy a system.twmrc) egy, a /etc/X11 könyvtárban lévő konfigurációs fájlra kellene utalni, amely egy hivatkozás (*link*) is lehet egy a /usr/X11/lib könyvtárban helyetfoglaló fájlra.

4.2. /usr/X386: az X Window Rendszer, 11. verzió, 5. kiadása az x86 platformokon

Ez a felépítmény (*hierarchia*) általában megegyezik a /usr/X11R6 hierarchiával; a /usr X11-es szimbolikus hivatkozásainak (*links*) a kívánt verziójú X Window Rendszerre kellene mutatniuk.

4.3. /usr/bin: a legtöbb felhasználói parancs

Ez a rendszeren található, futtatható parancsok elsődleges könyvtára.

/**usr**/**bin** — binárisok, amelyek nem szükségesek egyfelhasználós (*single-user*) módban

mh	parancsok az MH levélkezelő rendszer számára
X11	szimbolikus hivatkozás (*link*) a /usr/X11R6/bin könyvtárra

Mivel a burokprogram (*shell*) script értelmezői (meghívva a #!<elérési_út> -val a shell script első sorában) nem építenek az elérési útra (*path*), ezért előnyös szabványosítani az elhelyezkedésüket. A Bourne és a C-parancsértelmezők helye már rögzített a /bin könyvtárban, de a Perl, Python és a Tcl fordítók gyakran találhatók igen eltérő helyeken. A /usr/bin/perl, a /usr/bin/python és a /usr/bin/tcl programoknak a perl, python és a tcl burokprogram (*shell*) értelmezőkre kellene hivatkozniuk. Ezek lehetnek szimbolikus hivatkozások (*links*) az adott parancsértelmező fizikai helyére.

4.4. /usr/include: a szabványos include fájlok könyvtára

Ezen a helyen tároltak a rendszer általánosan használt include fájljai a C és a C++ programozási nyelvek számára.

/**usr**/**include** — include fájlok

X11	szimbolikus hivatkozás (*link*) a /usr/X11R6/include/X11 könyvtárra
bsd	a BSD kompatibilitás include fájljai (amennyiben szükségesek)
g++	a GNU C++ include fájljai

4.5. /usr/lib: a csomagok és a programozás programkönyvtárai

A `/usr/lib` object fájlokat, programkönyvtárakat, valamint olyan belső bináris állományokat tartalmaz, amelyek nem a felhasználók általi vagy a parancsértelmező scriptekbőli (*shell scripts*) közvetlen futtatására vannak tervezve.

Az alkalmazások használhatnak egy egyszerű könyvtárat a `/usr/lib` alatt. Amennyiben egy alkalmazás használ egy alkönyvtárat, valamennyi architektúra-függő adatát, amely kizárólag ezen alkalmazás által használt, ebben az alkönyvtárban kellene elhelyezni. Például a `perl5` nevű könyvtár a Perl 5 moduljainak és programkönyvtárainak tárolására szolgál.

A különféle architektúra-független, alkalmazás-specifikus (egyedileg jellemző), statikus (nem változó) fájlokat és alkönyvtárakat a `/usr/share` könyvtárban kellene elhelyezni.

Néhány futtatható parancs, mint például a `makewhatis` és a `sendmail` hagyományos helye is a `/usr/lib`. A `makewhatis` egy belső bináris állomány és egy, a binárisok számára fenntartott könyvtárban kellene elhelyezni (felhasználói hozzáférést kizárólag a `catman` számára engedélyezve). Az újabb `sendmail` binárisok esetében, ahol az alapértelmezett könyvtár a `/usr/sbin`, egy szimbolikus hivatkozásnak (*link*) kellene maradnia a `/usr/lib` könyvtárból. Továbbá, azokon a rendszereken, amelyeken az Smail programot használják, az Smail programnak a `/usr/sbin/smail` alatt kellene helyet foglalnia és egy `/usr/sbin/sendmail` nevü szimbolikus hivatkozásnak (*link*) is szükséges a megléte, amely a `/usr/sbin/smail`-re mutat.

Megkívánt, hogy a `/usr/lib/X11` szimbolikus hivatkozás (*link*) az alapértelmezett X terjesztés (*distribution*) `lib/X11` könyvtárára mutasson, amennyiben az X installált a rendszerre.

Megjegyzés: Az X Window Rendszer nem gépspecifikus adatait a `/usr/lib/X11` *könyvtárban kellene tárolni. Az olyan gépspecifikus fájlokat, mint az* `Xconfig` *vagy az* `XF86Config`*, a* `/etc/X11` *könyvtárban kellene tárolni. Ez vonatkozik az olyan konfigurációs fájlokra is, mint a* `system.twmrc`*, még akkor is, ha ez csak egy szimbolikus hivatkozás (*link*) egy globálisabb konfigurációs fájlra (amely esetleg a* `/usr/X11R6/lib/X11` *könyvtárban található).*

4.6. /usr/local: a helyi felépítmény (hierarchia)

A `/usr/local` hierarchiát a rendszeradminisztrátor használja, amikor helyileg installál egy szoftvert. A rendszer szoftver frissítése előtt szükséges lehet egy mentés is, a teljes felülírás által elvesztett adatok elkerülésére. Ez a hierarchia olyan programok és adatok számára használható fel, amelyek megoszthatók egy csoportnyi számítógép között, de nem találhatók meg a `/usr` hierarchiában.

/usr/local — a helyi felépítmény (hierarchia)

`bin`	helyi binárisok
`games`	a helyi játékok binárisai
`include`	helyi C fejléc (*header*) fájlok
`lib`	helyi programkönyvtárak
`sbin`	helyi rendszerbinárisok
`share`	helyi architektúra-független adatállományok
`src`	helyi forráskód

Egy FHS-engedékeny rendszer első installálása után ennek a könyvtárnak mindig üresnek kellene lennie. Nincs kivétel e szabály alól. Itt semmi másnak nem kellene lennie, csak az imént kilistázott könyvtárak csonkjainak.

A helyileg installált szoftverek helye inkább a `/usr/local` legyen, mint a `/usr`, hacsak az nem egy hozzáfűzés vagy frissítés egy, a `/usr` könyvtárban található programhoz.

Jegyezzük meg, hogy a `/` és a `/usr` könyvtárak felülírásra kerülnek egy rendszerfrissítés során (bár azt javasoljuk, hogy a `/etc` könyvtárban található adatokat a terjesztések (*disztribúciók*) ne írják felül a frissítés folyamatai alatt). Emiatt a helyi szoftvereket nem tanácsos a `/usr/local` könyvtáron kívül elhelyezni, legalábbis alapos ok nélkül nem.

4.7. /usr/sbin: nem alapvető, szabványos rendszerbinárisok

Ez a könyvtár kizárólag a rendszeradminisztrátor által használt, nem létfontosságú binárisokat tartalmazza. Az olyan rendszeradminisztrációs feladatok programjait, mint például a rendszer javítása, visszaállítása, a `/usr` becsatlakoztatása, vagy egyéb létfontosságú funkciók; inkább a `/sbin` könyvtárba kellene elhelyezni.

A `/usr/sbin` könyvtár tipikusan hálózati démonokat, tetszőleges nem létfontosságú adminisztrációs eszközöket tartalmaz, valamint bináris állományokat nem kritikus kiszolgáló (*server*) programokhoz.

Ezek a kiszolgáló programok használatosak, belépve az olyan System V állapotokra – ismertebben –, mint a „2. futási szint” (több felhasználós állapot = *multi-user state*) és a „3. futási szint” (hálózati állapot = *networked state*), vagy a BSD állapotra – ismertebben –, azaz a „több felhasználós módba” (multi-user mode). Ennél a pontnál teszi a rendszer elérhetővé a szolgáltatásait a felhasználók (például a nyomtató támogatás) és más gépek (például az NFS exportálások) részére.

A helyileg installált rendszeradminisztrációs programokat a `/usr/local/sbin` könyvtárba kellene elhelyezni.

4.8. /usr/share: architektúra-független adatok

/**usr**/**share** — architektúra-független adatok

`dict`	szó-listák
`doc`	különféle dokumentációk
`games`	statikus adatfájlok a `/usr/games` könyvtár adatainak számára
`info`	a GNU info rendszer elsődleges könyvtára
`locale`	helyi információk
`man`	online kézikönyvek
`nls`	anyanyelvi támogatás
`misc`	különféle architektúra-független adatok
`terminfo`	a terminfo adatbázis könyvtárai
`tmac`	a groff-val nem terjesztett troff makrofájlok
`zoneinfo`	időzóna információ és konfiguráció

A `/usr/share` hierarchia a csak olvasható (*read-only*), architektúra-független adatfájlokat tartalmazza. Az itt található adatok közül jónéhánynak az eredeti lelőhelye a `/usr` (`man`, `doc`) vagy a `/usr/lib` (`dict`, `terminfo`, `zoneinfo`) könyvtárban volt. Ez a könyvtárfa egy adott operációs rendszerű, tetszőleges architektúrával rendelkező platformú gépek közötti megosztásra szánt, ezért például egy i386-os, egy Alpha és egy PPC platformú számítógép is fenntarthat egyetlen központilag becsatlakoztatott (*mountolt*) `/usr/share` könyvtárat. Ám az mindenképpen megjegyzendő, hogy a `/usr/share` könyvtár általában nem arra szánt, hogy megosztott legyen eltérő operációs rendszerű vagy különböző kiadású, de azonos típusú operációs rendszerű számítógépek között.

Bármely programot vagy csomagot, amelyben olyan adatok vannak, amelyek nem tartalmaznak vagy nem követelnek működésükhöz változó tartalmú adatokat, a `/usr/share` könyvtárban kellene tárolni (vagy a `/usr/local/share`-ban, ha helyileg installáltuk azt). E célra a `/usr/share` könyvtár ajánlott.

Fontos tudnunk, hogy a Linux pillanatnyilag a DBM-formátumú adatbázisokat használja. Mindaddig, amíg ezek az adatbázisok nem lesznek architektúra-függetlenek, engedélyezett számunkra a `/usr/share` könyvtár – számítva egy esetleges architektúra-független DB 2.0 választására.

A `/usr/share/games` könyvtában kizárólag a játékprogramok statikus adatfájljait kellene tárolni. Bármilyen ettől eltérő tipusú fájlt, például az eredménylista fájlokat, a játékok időleges naplóállományait, és más egyéb fájlokat inkább a `/var/games` könyvtár alatt kellene tárolni.

Érdemes az alkalmazás-specifikus, architektúra-független könyvtárokat is ezen alkönyvtárban elhelyezni (ezek közé tartozhat többek között a `groff`, a `perl`, a `ghostscript`, a `texmf` és a `kbd` – a Linux alatt – programok könyvtárai vagy a `syscons` – a BSD rendszer alatt). E könyvtárak helye lehet a `/usr/lib` könyvtárban is, a visszafelé való kompatibilitás biztosítására. (Ezt a terjesztő (*disztribútor*) saját belátása szerint alkalmazhatja.) Hasonképpen a `/usr/lib/games` könyvtárfa adatai is tárolhatók a `/usr/share/games` könyvtárban, ha a terjesztő (*disztribútor*) néhány játék adatfájlját itt kívánja elhelyezni.

4.8.1. /usr/share/dict: szó listák

A /usr/share/dict könyvtár számára megkívánt fájlok:

{ words }

Hagyományosan ez a könyvtár csak az angol `words` fájlt tartalmazza, amelyet a `look(1)` parancs vagy különféle helyesírási programok használnak. A `words` fájlt használhatják az amerikai vagy a brit helyesírás programok is. Azok a gépek, amelyek számára szükséges, mindkét fájlra (a `/usr/share/american-english` vagy a `/usr/share/british-english` fájlra) készíthetnek egy-egy hivatkozást (*link*).

Más nyelvek szavainak listájai is használhatók, kicserélve az english szót az adott nyelv szavára. (Például `/usr/share/dict/french`, `/usr/share/dict/danish`, stb.) Ezeknek a fájloknak, amennyiben lehetséges azzal ISO 8859 karakter készlettel kellene rendelkezniük, amely megfelel az adott nyelv számára; ha lehetséges, akkor a Latin 1 (ISO 8859-1) karakterkészletet használjuk, bár ez sajnos gyakran nem lehetséges.

Más szó listákat, mint például a `web2` „szótár”, szintén ebben az alkönyvtárban kellene elhelyeznünk.

BEGIN ÉRTELMEZÉS

Az oka annak, hogy ebben a könyvtárban kizárólag csak szó lista fájlok találhatók az, hogy egyedül ezek a szokványos fájlok a helyesírásellenőrző programokhoz.

END ÉRTELMEZÉS

4.8.2. /usr/share/man: kézikönyvlapok

Ez a szakasz a /usr/share/man könyvtárban található kézikönyvlapok rendszeren belüli felépítési szerkezetét részletezi. A témával kapcsolatban a /var/cache/man könyvtárral foglalkozó részben is olvashatunk.

A kézikönyvlapok a <kézikönyvlap_könyvtár>/<elhelyezkedés>/kézikönyvlap<fejezet>/ <architektúra> szerkezet szerint épülnek fel, amely szerekezetet az alábbiakban értelmezzük.

<**kézikönyv_könyvtár**>/<**elhelyezkedés**> — kézikönyvlapok hierarchiája

man1	felhasználói programok
man2	rendszerhívások
man3	programkönyvtár hívások
man4	speciális fájlok
man5	fájl formátumok
man6	játékok
man7	egyebek
man8	rendszeradminisztráció

A rendszer elsődleges <kézikönyv_könyvtára> a /usr/share/man könyvtár, amely a / és a /usr fájlrendszerek parancsainak és adatainak kezelési információit tartalmazza. Nyilvánvaló, hogy a / fájlrendszerben nincsenek kézikönyvlapok, mivel ezekre sem a bootolási idő alatt, sem a rendszerrel kapcsolatos vészhelyzetekben sincs szükség.

A <fejezet> a kezelési kézikönyv fejezetét írja le.

Intézkedéseket kell tenni annak érdekében, hogy a /usr/share/man struktúra támogassa az eltérő vagy többnyelvű kézikönyvlapokat. Ezeket az intézkedéseket a kézikönyvlapok tárolási nyilvántartásában és a lapok hivatkozási helyére vonatkozóan kell foganatosítani. Lényeges tényezők közé tartozik a kézikönyvlapok nyelve (beleértve a földrajzon alapuló eltéréseket is) és azok karakter kódkészlete.

A /usr/share/man könyvtárban található, eltérő nyelvű könyvtárak elnevezése a POSIX 1003.1 szabvány E mellékletén alapul, amely leírja a helyi azonosító karaktersorozatot (ez a legszélesebben elfogadott módszer a kulturális környezetek leírására). Az <elhelyezkedés> karaktersorozata a következőkből áll:

```
<nyelv>[<terület>][.<karakterkészlet>][,verzió]
```

A <nyelv> mezőt az ISO 639 alapján kellene megadni, amely egy kódrendszer a nyelvek neveinek ábrázolására. Ez két karakter hosszú lehet és csak nagybetűkkel adható meg.

A <terület> mezőnek az ISO 3166 szabvány kétbetűs kódjának kellene lenni (ez – egy – az országok ábrázolásával kapcsolatos meghatározás), amennyiben ez lehetséges. Ez a legtöbbek számára jól ismert kétbetűs kód, amelyet az e-mail címek[1] országkódjában is használnak. Ennek szintén kétbetűsnek kellene lennie és csak nagybetűk használhatók.

A <karakterkészlet> mező ábrázolja a szabványos karakterkészletet. Ha a <karakterkészlet> mező csak számokkal van meghatározva, a szám jeleníti meg a karakterkészlet nemzetközi szabványának

1. Az Egyesült Királyság a legfőbb kivétel e szabály alól, mivel az ISO 3166-ban a „GB" található meg, bár a legtöbb email címben a „UK" a használatos.

számértékét. Kívánatos, hogy ez az érték számokkal legyen ábrázolva; ha ez lehetséges (főként ISO szabványokkal), ne tartalmazzon további írásjeleket, és a betűket kisbetűkkel írjuk.

A jellemző `<verzió>` paraméterének meghatározása a `<karakterkészlet>` mező után helyezkedik el, vesszővel határolva. Ez használatos az eltérő kultúrák közötti különbségtételre (például a szótár szerint rendszerezett, a rendszerorientáltabb gyűjteményi rendszerezéssel szemben). Ez a szabvány nem javasolja a `<verzió>` mező használatát, hacsak az nem szükséges.

Azok a rendszerek, amelyek egyedi nyelvet és kódkészletet használnak, elhagyhatják az `<elhelyezkedés>` almezőt és tárolhatják a `<kézikönyv_könyvtár>`-ban valamennyi kézikönyvlapot. Például egy rendszer, amelynek csak angol (English) nyelvű és ASCII kódolású kézikönyvlapjai vannak, tarthatja valamennyi kézikönyvlapját (a `kézikönyv<fejezet>` könyvtárakat) közvetlenül a `/usr/share/man` könyvtárban (valójában ez a hagyományos eset és elrendezés).

Azok az országok, amelyekhez létezik elfogadott szabványos karakterkészlet, természetesen elhagyhatják a `<karakterkészlet>` mezőt, bár ennek megtartása erősen javallott, különösen azon országok esetében, amelyekben több eltérő szabvány is konkurál.

Néhány példa:

Nyelv	Terület	Karakterkészlet	Könyvár
angol	—	ASCII	/usr/share/man/en
angol	Egyesült Királyság	ASCII	/usr/share/man/en_GB
angol	Egyesült Államok	ASCII	/usr/share/man/en_US
francia	Kanada	ISO-8859-1	/usr/share/man/fr_CA
francia	Franciaország	ISO-8859-1	/usr/share/man/fr_FR
német	Németország	ISO 646	/usr/share/man/de_DE.646
német	Németország	ISO 6937	/usr/share/man/de_DE.6937
német	Németország	ISO 8859-1	/usr/share/man/de_DE.88591
német	Svájc	ISO 646	/usr/share/man/de_CH.646
japán	Japán	JIS	/usr/share/man/ja_JP.jis
japán	Japán	SJIS	/usr/share/man/ja_JP.sjis
japán	Japán	UJIS (vagy EUC-J)	/usr/share/man/ja_JP.ujis

Hasonló elrendezés szükséges az architektúra-független kézikönyvlapok számára is (ilyenek például az eszközmeghajtó programok dokumentációi vagy az alacsony szintű rendszeradminisztrációs parancsok). Ezeket az `<arch>` könyvtárban kellene elhelyezni, a megfelelő `kézikönyv<fejezet>` könyvtárban (például a ctrlaltdel (8) i386 architektúrájú kézikönyvlap a `/usr/share/man/<terület>/man8/i386/` könyvtárba tehető `ctrlaltdel.8` néven).

A `/usr/local` könyvtár parancsainak és adatainak kézikönyvlapjai a `/usr/local/man` könyvtárban tárolhatók. Az X11R6 kézikönyvlapjai a `/usr/X11R6/man` könyvtár alatt tárolhatók. Az ebben a könyvtárban található kézikönyvlapok hierarchiájának, a `/usr/share/man` könyvtárral megegyező stuktúrájúnak kellene lennie. Az üres alkönyvtárak elhagyhatók a kézikönyvlapok hierarchiájából. (Például, ha a `/usr/local/man` könyvtár 4. fejezetében (Eszközmeghajtók) nem található egyetlen kézikönyvlap sem, akkor a `/usr/local/man/man4` könyvtár elhagyható.)

A cat lapok fejezetei (`cat<fejezet>`) tartalmazzák azon megformázott kézikönyvlap bejegyzéseket, amelyek megtalálhatók a `<kézikönyv_könyvtár>/<elhelyezkedés>` alkönyvtárakban is, de ezeknek nem szükséges a terjesztése, szemben az nroff forrás kézikönyvlapokkal.

A kézikönyvlapok 1-től 8-ig számozott fejezetei tradicionálisan definiáltak. Általában egy adott nevű kézikönyvlap fájl a `.<fejezet>`-számot, a tartalmára világosan utaló neve után viseli.

Néhány nagyobb alkalmazás-specifikus kézikönyvlap csoportnak lehet még további rag, szótoldalék hozzáfűzés a csomag kézikönyvlapjainak nevében. Például az MH levélkezelő rendszer valamennyi kézikönyvlapjai lapjának nevéhez hozzá kellene fűzni az `mh` ragot. Valamennyi X Window Rendszerrel kapcsolatos kézikönyvlap nevéhez hozzá kellene fűzni egy `x`-et.

A gyakorlatban a különféle nyelvű kézikönyvlapok a `/usr/share/man` megfelelő alkönyvtáraiban vannak, amely alkalmazható más kézikönyvlap hierarchiákra – például a `/usr/local/man` vagy a `/usr/X11R6/man` könyvtárak esetében – is (a szabvány e része alkalmazott a későbbiekben az opcionálisan használható `/var/cache/man` stuktúra esetében is).

Következzen valamennyi fejezet rövid leirása:

- `man1`: Felhasználói parancsok
  Ebbe a fejezetbe azok a kézikönyvlapok tartoznak, amelyek a nyilvánosan hozzáférhető parancsokat írják le. Itt van elhelyezve a legtöbb olyan dokumentáció, amelyre a felhasználóknak szükséges lehet.
- `man2`: Rendszer hívások
  Ez a fejezet írja le valamennyi rendszerhívást, amelyek a kernel működéséhez szükségesek.
- `man3`: Programkönyvtár függvények és alrutinok
  A harmadik fejezet azokat a programkönyvtár rutinokat írja le, amelyek nem közvetlenül hívják a kernel szolgáltatásait. A programozók számára valójában csak a 2. és a 3. fejezet fontos igazán.
- `man4`: Speciális fájlok
  A 4. fejezet a speciális fájlokat írja le, a kapcsolódó eszközmeghajtó funkciókkal és a rendszeren elérhető hálózati támogatással együtt. Általában itt található a `/dev` könyvtár eszközmeghajtó fájljainak és a hálózati protokollt támogató kernel illesztőfelületének (*interface*) leírása.
- `man5`: Fájl formátumok
  Az igen sok és első ránézésre nem mindig megérthető adatfájlok formátumainak leírása található az 5. fejezetben. Ez a fejezet tartalmazza a különféle include fájlokat, a programok kimeneti állományait és a rendszerfájlokat.
- `man6`: Játékok
  Ez a fejezet dokumentálja a játékokat, a demókat és általában a magától értetődő programokat. A vélemények eltérőek lehetnek abban a tekintetben, hogy ezek mennyire nélkülözhetetlenek.
- `man7`: Egyebek
  A 7. fejezethez azok a kézikönyvlapok lettek kijelölve, amelyeket nehéz osztályozni. A troff és más szövegformázó makrocsomagok is itt találhatóak.
- `man8`: Rendszer adminisztráció
  A rendszeradminisztrátor által a rendszer működéséhez és fenntartásához használt programok dokumentációi találhatók ebben a fejezetben. E programok közül némely, esetenként a normál felhasználók számára is hasznos lehet.

4.8.3. /usr/share/misc: egyéb, architektúra-független adatok

Ez a könyvtár tartalmazza azokat az architektúra-független fájlokat, amelyek nem kívánnak külön alkönyvtárakat a `/usr/share` alatt. Ez a könyvtár mindenképpen megkívánt a `/usr/share` könyvtár alatt.

A következő fájlokat kellene elhelyezni a `/usr/share/misc` könyvtár alatt, amennyiben jelen vannak a rendszeren.

```
{ ascii, magic, termcap, termcap.db }
```

Más (alkalmazás-specifikus) fájlok is feltűnhetnek itt, bár a terjesztő (*disztribútor*) belátására van bízva, hogy ide helyezi-e azokat vagy a `/usr/lib` könyvtárba. Néhány fájl ezek közül:

`{ airport, birthtoken, eqnchar, getopt, gprof.callg, gprof.flat, inter.phone, ipfw.samp.filters, ipfw.samp.scripts, keycap.pcvt, mail.help, mail.tildehelp, man.template, map3270, mdoc.template, more.help, na.phone, nslookup.help, operator, scsi_modes, sendmail.hf, style, units.lib, vgrindefs, vgrindefs.db, zipcodes }`

4.9. /usr/src: forráskód

Bármely nem-helyi forráskódot ebben az alkönyvtárban kellene elhelyezni.

5. A /var hierarchia

/**var** — változó adatok

`account`	folyamatszámlázási naplófájlok (amennyiben támogatott)
`cache`	alkalmazás cache adatok betöltő statikus fájljai
`crash`	rendszerösszeomlási fájlok tárolókönyvtára
`games`	a játékok változó adatfájljai
`lock`	lezárófájlok
`log`	naplófájlok és könyvtárak
`mail`	felhasználói levelesláda fájlok
`opt`	a `/opt` könyvtár változó adatfájljai
`run`	az éppen futó folymatokhoz tartozó fájlok
`spool`	Alkalmazások várakozási sorainak (*spool*) fájljai
`state`	változó állapot információk
`tmp`	két bootolás között megőrzött ideiglenes fájlok
`yp`	a Hálózati Információ Szolgálat (Network Information Servise – NIS) adatbázis fájljai

A `/var` könyvtár tartalmazza a változó tartalmú és méretű adatfájlokat. A hierarchia magában foglalja a várakozási sorok (*spool*) könyvtárait és fájljait, az adminisztrációs és a naplózási adatokat, valamint az átmeneti, ideiglenes fájlokat is.

A `/var` némely része nem osztható meg különböző számítógépek között. Például a `/var/log`, a `/var/lock` és a `/var/run` könyvtárak. Néhány része viszont megosztható, nevezetesen a `/var/mail`, a `/var/cache/man`, a `/var/cache/fonts` és a `/var/spool/news` könyvtárak.

A `/var` hierarchia alkalmazása tette lehetővé azt, hogy a `/usr` struktúra becsatlakoztatható (*mount*) csak olvasható (*read-only*) módon. Minden, ami előzőleg a `/usr`-ben volt és módosításra került a rendszer működése alatt (szemben az installációval és a programok fenntartásával), ezután a `/var`-ban kell lennie.

Az olyan esetekben, amikor a `/var` nincs külön partíción, gyakran jobb megoldás kimozgatni a root partícióról és áttenni a `/usr` fájlrendszer partíciójára. (Ezzel csökkenthetjük a root partíció méretét vagy növelhető a leapadt szabad terület a root partíción.) Bárhogyis, de a `/var` könyvtárat nem kellene hivatkozásként (*link*) beépíteni a `/usr` struktúrába, mivel ez a `/usr` elkülönítését okozza, a `/var` struktúrát is bonyolítja és valószínűleg elnevezésbeli hibákat fog előidézni. Ehelyett inkább lánculjuk (*link*) a `/var` struktúrát a `/usr/var` könyvtárra.

Az alkalmazásoknak általában nem kellene könyvtárakat hozzáadni a `/var` struktúra legfelső szintjéhez. Csak olyan könyvtárakat kellene hozzáadni a hierarchiához, amelyek működése a rendszer egészét érinti, és javasolt ennek előzetes egyeztetése az FHS levelező listával.

A `/var` felépítésnek tartalmaznia kell a `cache`, `lock`, `log`, `run`, `spool`, `state` és a `tmp` könyvtárakat – valamennyi terjesztés (*disztribúció*) használja is ezeket a könyvtárakat –, valamint szintén szükséges tartalmaznia az `account`, `crash`, `games`, `mail` és az `yp` könyvtárakat és használnia is szükséges a terjesztésnek ezeket a könyvtárakat, amennyiben a megfelelő alkalmazás vagy tulajdonság a terjesztés által ellátott.

Az FSSTND előző verziója a `/var/lib` hierarchiát is tartalmazta. Az ehhez kapcsolódó további információkat lásd a `/var/state` könyvtárhoz tartozó résznél.

Számos olyan könyvtár fenntartott, amelyeket nem kellene használnia némely újabb alkalmazásnak, amióta ezek ellentétbe kerülhetnek a történelmi és/vagy a helyi gyakorlattal.

Ezek a következők:

`/var/backups`

`/var/cron`

`/var/lib`

`/var/local`

`/var/msgs`

`/var/preserve`

5.1. /var/account: folyamat számlázási naplófájlok (amennyiben ez támogatott)

Ebben a könyvtárban vannak az éppen aktív folyamatok számlázási naplófájljai és az olyan összetett folyamatok által használt adatok, mint például amit néhány UNIX-típusú rendszeren a `lastcomm` vagy az `sa` parancs használ.

5.2. /var/cache: alkalmazások cache adatai

/**var**/**cache** — Cache könyvtárak

`fonts`	a helyileg generált fontok
`man`	a helyileg formázott kézikönyvlapok
`www`	www proxy vagy cache adatok
`<csomag>`	csomagspecifikus cache adatok

A `/var/cache` könyvtár szánt az alkalmazások adatai cachelésének céljára (olyanoknak, mint például az I/O időfelhasználás eredményeként helyi szinten generált adatok vagy a különböző számítások). Az alkalmazásoknak képeseknek kell lenni újra létrehozni vagy visszaállítani az adataikat. A `/var/spool` könyvtártól eltérően a cachelt fájlok adatvesztés nélkül letörölhetők, és az adatoknak valósnak kellene maradniuk az adott alkalmazás két eltérő idejű használata között, illetve két bootolás között is.

A `/var/cache` könyvtárban elhelyezett adatoknak vagy az adott alkalmazásra jellemző módon kellene elenyészni, vagy a rendszeradminisztrátor is foganatosíthatja ezt; esetleg mindkettő. Az adott alkalmazásnak pedig mindig képesnek kell lennie a kézzel törölt adatok visszaállítására (hiszen a szabad lemezterületek gyorsan betelnek). Nincs egyéb követelmény a cache könyvtárak adatformátumával kapcsolatban.

BEGIN ÉRTELMEZÉS

A több, különálló könyvtár lehetősége az adatok cachelése számára, engedélyezi a rendszeradminisztrátor részére azt, hogy különböző lemezkezelési és adatmentési irányvonalat alakíthasson ki a /var könyvtáraira nézve.

END ÉRTELMEZÉS

5.2.1. /var/cache/fonts: helyileg generált fontok

A /var/cach/fonts könyvtárat kellene használni a dinamikusan generált fontok tárolására. Főként a /var/cache/fonts/pk könyvtárban tárólódnak azok a fontok, amelyek a MakeTeXPK által generálódnak automatikusan.

Készíteni kellene egy hivatkozást (*link*) a /usr/lib/texmf/fonts/tmp könyvtárról a /var/cache/fonts könyvtárra. E hivatkozás segítségével lehetségessé válik a felhasználó számára, hogy egyszerűen használják a /usr/lib/texmf/fonts/tfm elérési utat, amikor megváltoztatják a TEXFONTS környezeti változójuk értékét. (Ez ugyanis az alapértelmezett elérési út Karl Berry TeX segédcsomagjainak, amely az ftp.cs.umb.edu:/pub/tex[2] címen beszerezhető. Amennyiben egy másik TeX terjesztés használt, akkor szükséges lehet létrehozni egy hivatkozást (*link*) az adott font könyvtárból a /var/cache/fonts könyvtárra.)

A MakeTeXPK, amely a dvipsk programhoz tartozik, a .pk fájljait a fonts/pk/<eszközmeghajtó>/<fontnév> alá fogja elhelyezni (például, fonts/pk/CanonCX/cmr10.300pk) A .pk fájlok periodikusan kerülnek megtisztításra a /var/cache/fonts fában, vagy átmozgathatók a /usr/lib/texmf fa alá. Ha automatikus .mf vagy .tfm gererátor van használatban, akkor ezeknek a /var/cache/fonts megfelelő – mf vagy tfm – alkönyvtáraiba kellene tenniük az adataikat.

Egyéb dinamikusan létrehozott fontok is elhelyezhetőek ebben a fában, a /var/cache/fonts könyvtár megfelelően elnevezett alkönyvtárai alatt.

5.2.2. /var/cache/man: a helyileg formázott kézikönyvlapok (szabadon választható)

A /var/cache/man a könyvtár azon gépek részére lát el egy szabványos müködést, amelyek a /usr hierarchiát csak olvasható (*read-only*) partícióról használják, de engedélyezni kívánják a helyileg formázott kézikönyvlapok cachelését. Azok a számítógépek, amelyek a /usr fát írhatóan csatlakoztatják be (például egyfelhasználós installálás esetén) választhatják azt is, hogy nem használják a /var/cache/man könyvtárat és ez esetben írhatják közvetlenül a formázott kézikönyvlapokat a cat<fejezet>-en keresztül a /usr/share/man könyvtárba. Javasoljuk, hogy a legtöbb számítógépen a következő példák közül használjuk valamelyiket:

- Készítsük el a gépen található valamennyi – még nem formázott – kézikönyvlap előformázását.
- Ne engedélyezzük a formázott kézikönyvlapok cachelését és követeljük meg valamennyi kézikönyvlap minden egyes használat előtti formázását.
- Engedélyezzük a formázott kézikönyvlapok helyi cachelését a /var/cache/man könyvtárban.

2. Az ok, amiért Karl Berry eszközcsomagját említettük meg az, hogy ezek az eszközök ténylegesen szabványosak a TeX UNIX alapú installációi számára és széles körben használtak a szabad UNIX közösségen belül.

A /var/cache/man struktúrának tükröznie kell a többszörös kézikönyvlapok valós hierarchiáját, de a különböző nyelvek támogatásának lehetőségét is.

Egy adott, még nem formázott kézikönyvlap normális esetben az <elérési_út>/man/<elhelyezkedés>/man<fejezet> könyvtárban tűnhet fel, a formázott kézikönyvlapok helye pedig a /var/cache/man/<cat_elérési_út>/<elhelyezkedés>/cat<fejezet>, ahol a <cat_elérési_út> az <elérési_út> értékéből származik, elhagyva bármilyen vezető usr és/vagy az utána következő share elérési út részletet.(Jegyezzük meg, hogy az <elhelyezés> összetevő hiányozhat.)

Például, a /usr/share/man/man1/ls.1 formázott lehet a /var/cache/man/cat1/ls.1-be és a /usr/X11R6/man/<elhelyezkedés>/man3/XtClass.3x a /var/cache/man/X11R6/<elhelyezkedés>/cat3/XtClass.3x-be.

A /var/cache/man könyvtárba írott kézikönyvlapok végül átkerülnek a forrás man hierarchia megfelelő előformázási könyvtáraiba vagy elenyésznek éppúgy, mint a forrás man hierachia formázott kézikönyvlapjai, amennyiben egy adott időperióduson belül nem történik hozzáférés.

Ha az előformázott kézikönyvlapok egy csak-olvasható rendszerű médiáról érkeznek (például CD-ROM), e lapokat a forrás man hierarchiába kellene installálni (például /usr/share/man/cat<fejezet>). A /var/cache/man könyvtár a formázott kézikönyvlapok számára van fentartva írható cachelésre.

BEGIN ÉRTELMEZÉS

A szabvány 1.2 verziója a /var/catman könyvtárat határozta meg e hierarchia számára. Az elérési út a /var/cache könyvtárra változott, amely jobban tükrözi a formázott kézikönyvlapok dinamikus sajátosságait. A könyvtár neve man-ra változott, hogy hangsúlyozni engedje a felépítést, amelybe beletartoznak az utólag (nem a „cat" által) feldolgozott formátumok (iyenek például a Postscript, HTML vagy a DVI).

END ÉRTELMEZÉS

5.3. /var/crash: Rendszerösszeomlási tárkiírások (amennyiben támogatott)

Ez a könyvtár tárolja a rendszerösszeomláskor a winchesterre kiírt memória tartalmát. E szabvány ezen verziójának kiadási dátumakor a Linux nem támogatta a rendszerösszeomlási tárkiírásokat.

5.4. /var/games: A játékok változó adatai

Bármely, a /usr könyvtárban lévő, játékkal kapcsolatos, módosúló adatot ebben a könyvtárban kellene elhelyezni. A /var/games könyvtárban azokat a változó tartalmú adatokat kellene tárolni, amelyek helye korábban a /usr könyvtárban volt. A statikus adatoknak, mint például a help szövegek, szint leírások, stb., valahol máshol kellene (például a /usr/share/games könyvtárban) maradni.

Ami a /var/state könyvtárat illeti, a játékok változó tartalmú adatait a /var/lib könyvtárban lehet elhelyezni, amely csak egy átmeneti, nem egészen megfelelő megoldás. Ez az engedmény a szabvány későbbi verzióiban meg fog szűnni.

BEGIN ÉRTELMEZÉS

A /var/games inkább a saját hierarchiáját adta meg, mintsemhogy hagyja azt egyesítetten a régi /var/lib könyvtárral, amely a szabvány 1.2 verziójában volt. Ez az elválasztás helyi ellenőrzés alatt hagyja a mentési stratégiákat, az engedélyeket és a lemezhasználatot csakúgy, mint a gépen belüli megosztás engedélyezését, ezen kívül a `/var/state` könyvtárban lévő rendetlenséget is csökkenti. A `/var/games` elérési út a BSD rendszerben használt hagyományosan.

END ÉRTELMEZÉS

5.5. /var/lock: Lezáró (lock) fájlok

A lezáró fájlokat a `/var/lock` könyvtár struktúrában kellene tárolni.

Az eszközmeghajtók lezáró fájljait éppúgy, mint a soros eszköz lezáró fájljait – amelyek eredetileg vagy a `/usr/spool/locks`, vagy a `/usr/spool/uucp` könyvtárban voltak fellelhetők –, most a `/var/lock` könyvtárban kellene tárolni. Az elnevezési megkötések, amelyeket be kellene tartani: `LCK..` és ezt követi az eszközmeghajtó alapneve. Például, a `/dev/cua0` lezárására az `LCK..cua0` fájl jöhet létre.

Az eszközmeghajtó fájlok lezárására használt fájlnevek formátumának HDB UUCP lezáró fájl formátumúnak kell lennie. A HDB formátum tárolja a folyamat (*processz*) azonosítót (PID), 10 byte-os ASCII decimális számként, egy ezt követő új sorral. Például, ha az 1230-as folyamat (*processz*) fenntart egy lezáró fájlt, annak 11 karaktert kellene tartalmaznia: szóköz, szóköz, szóköz, szóköz, szóköz, szóköz, egy egyest, egy kettest, egy hármast, egy nullát és egy újsor karaktert.

Majd ezután, ha bármi is használni kívánja a `/dev/cua0`-t, elolvassa a lezárófájlt, majd ennek megfelelően cselekszik (valamennyi lezárófájlnak mindenki számára olvashatónak kellene lenni a `/var/lock` könyvtárban).

5.6. /var/log: Naplófájlok és könyvtárak

Ez a könyvtár különféle naplófájlokat tartalmaz. A legtöbb naplófájlt vagy ebbe a könyvtárba kellene írni, vagy e könyvtár egy megfelelő alkönyvtárába.

`lastlog`	rögzíti valamennyi felhasználó utolsó bejelentkezését
`messages`	a `syslogd` démontól származó rendszerüzenetek
`wtmp`	rögzíti valamennyi felhasználó be- és kijelentkezését

5.7. /var/mail: Felhasználói levelesláda fájlok

Ez a könyvtár tartalmazza a felhasználói levelesláda fájlokat, amelyek szabványos UNIX levelesláda formátumban tároltak.

BEGIN ÉRTELMEZÉS

Ez a könyvtár a `/var/spool/mail` könyvtárból került át azon célból, hogy kapcsolatba hozza az FHS-el csaknem valamennyi UNIX implementációt. Ez a változtatás igen fontos a belső kezelhetőség szempontjából, mióta a `/var/mail` könyvtár igen gyakran megosztott egyszerre több számítógép és UNIX implementáció között (dacára az NFS lezárási következményeinek).

END ÉRTELMEZÉS

5.8. /var/opt: A /opt könyvtár változó tartalmú adatai

A változó adatokat `/var/opt/<csomag>` könyvtárba kellene installálni, ahol a `<csomag>` annak a fának a neve a `/opt` könyvtárban, amelyben egy utólagos szoftver csomagja tárolt. Kivételt képez ez alól az az eset, amikor az előbbi megoldást egy `/etc` könyvtár alatti fájl helyettesíti. Nincs előírt struktúra a `/var/opt/<csomag>` belső elrendezésére.

BEGIN ÉRTELMEZÉS

Nézz utána a `/opt` értelmezésének.

END ÉRTELMEZÉS

5.9. /var/run: Futási idő alatt változó fáljok

Ez a könyvtár tartalmazza azokat a rendszerinformációs fájlokat, amelyek leírják a rendszert, amióta az bootolt. E könyvtár fájljait meg kellene tisztítani (törölni vagy csak megcsonkítani) a bootolás folyamatának kezdetekor.

A folyamat azonosító (PID) fájlokat, amelyek eredetileg a `/etc` könyvtárban voltak, a `/var/run` könyvtárban kellene lehelyezni. A PID fájlok elnevezési konvenciója: `<program-név>.pid`. Például a `cron` PID fájljának neve: `/var/run/cron.pid`.

A PID fájlok belső formátuma változatlan maradt. A fájl tartalmának továbbra is az ASCII kódolású, decimális folyamat azonosítóból (*process identifier*) kellene állnia, valamint az ezt követő újsor (*newline*) karakterből. Például, ha a `cron` folyamat (*process*) száma a 25, a `/var/run/cron.pid` fájl három karaktert tartalmazhat: egy kettest, egy ötöst és egy új sor karaktert.

A programoknak, amelyek olvassák a PID fájlokat, némileg hajlékonynak kellene lenniük abban a tekintetben, hogy mit fogadnak el (például semmibe kellene venniük az extra whitespace karaktert, a vezető nullákat, a követő újsor karakter hiányát vagy a PID fájlban lévő további sorokat). A programoknak, amelyek a PID fájlokat hozzák létre egy egyszerű meghatározást kellene elhelyezni a fentiek tekintetében.

Az `utmp` fájlt, amely azt tárolja, hogy kik használják jelen pillanatban a rendszert, szintén ebben a könyvtárban szükséges elhelyezni.

Az átmenő UNIX-domain socketeket karbantartó programokat szintén e könyvtárban kellene elhelyezni.

5.10. /var/spool: Alkalmazások várakozó sorainak (spool) adatai

/var/spool — A várakozási sorok (*spool*) könyvtárai

`lpd`	A nyomtató várakozási sorának könyvtára
`mqueue`	a kimenő levelek várakozósora
`news`	híranyag várakozási sorának könyvtára
`rwho`	az rwhod démon fájljai
`smail`	az smail várakozási sorainak könyvtárai
`uucp`	az UUCP várakozási sorának könyvtára

A `/var/spool` könyvtár tartalmazza azokat az adatokat, amelyek valamiféle későbbi folyamatra várakoznak. A `/var/spool` könyvtárban található adatok egy, a jövőben végrehajtandó munkát reprezentálnak (amely egy program, felhasználó vagy az adminisztrátor által lesz végrehajtva); az adatok gyakran törlésre kerülnek, miután folyamatuk befejeződött.

Az UUCP lezáró (lock) fájlokat a `/var/lock` könyvtárban kell elhelyezni. Lásd a `/var/lock` könyvtárral foglalkozó részt.

5.10.1. /var/spool/lpd: A sornyomtató démon nyomtatási várakozó sora

/**var**/**spool**/**lpd** — A nyomtatói várakozási sorának könyvtára

└── `<nyomtató>`	Speciális nyomtató göngyölegek (*spools*) adatai (opcionális)

Az `lpd` lezáró fálját – `lpd.lock` – a `/var/spool/lp` könyvtárban kellene elhelyezni. Javasolt, hogy valamennyi nyomtató lezáró fájlja a várakozási sor (*spool*) könyvtárában kerüljön elhelyezésre, nevében meghatározva a nyomtatót és a lock szót.

5.10.2. /var/spool/rwhod: Az rwhod démon fájljai

Ez a könyvtár tárolja az `rwhod` információkat más rendszerek számára a helyi hálózaton.

BEGIN ÉRTELMEZÉS

Némely BSD verzió a `/var/rwhod` könyvtárban tárolja ezeket az adatokat. Megadva ennek a történelmi elhelyezését a `/var/spool` könyvtárban, más rendszerek hozzávetőleges megfeleltetése a göngyölített (*spooled*) adatok definícióhoz, sokkal megfelelőbbnek tartott az elhelyezés szempontjából.

END ÉRTELMEZÉS

5.11. /var/state: Változó állapot információk

/**var**/**state** — Változó állapot információk

├── `<editor>`	Szövegszerkesztői (*editor*) fájlok biztonsági mentései és állapot fáljai
├── `misc`	Különféle állapot adatok
├── `xdm`	X megjelenítés kezelő változó adatok
├── `<pkgtool>`	A csomagolási támogatás fájljai
└── `<csomag>`	Állapot adatok a csomagok és az alrendszerek számára

Ez a hierarchia tartalmazza egy adott alkalmazáshoz vagy a rendszerhez tartozó állapot információkat. Az állapot információk olyan adatok, amelyeket a programok változtatnak, amíg azok futnak és egy adott számítógéphez tartoznak. A felhasználóknak lehetőleg ne legyen szükségük arra, hogy változtassák a `/var/run` könyvtárban ezeknek a fájloknak a tartalmát, hogy ezen keresztül konfigurálják egy csomag működését.

Az állapot információk szokták általában fenntartani egy alkalmazás (vagy egy belsőleg összefüggő alkalmazáscsoport) körülményeit megidézéseik között, illetve ugyanazon alkalmazásra való különböző hivatkozások között. Az állapot információknak általában érvényeseknek kellene maradni a bootolási folyamat után is, nem kellene naplózniuk a kimenetet és nem kellene adatokat göngyölíteniük.

Egy alkalmazásnak (vagy egy belsőleg összefüggő alkalmazáscsoportnak) egy alkönyvtárat kellene használni a `/var/state` könyvtárban saját adataik számára. Egy mindenképpen szükséges alkönyvtár van, és ez a `/var/state/misc`. A `/var/state/misc` könyvtár olyan állapot információk részére szánt, amelyeknek nincs szükségük külön alkönyvtárra; további könyvtáraknak csak abban az esetben kellene itt jelen lenniük, amennyiben a kérdéses alkalmazást tartalmazza az adott terjesztés (*distribution*).

A `/var/state/<name>` az a hely, amelyet valamennyi terjesztésnek (*distribution*) használnia kellene a csomagolás támogatására. Az eltérő terjesztések természetesen használhatnak eltérő elnevezéseket.

E szabvány előző kiadásai a `/var/lib` elnevezést használták erre a hierarchiára. A `/var/lib` használata nem javasolt, bár használható párhuzamosan a mindenképpen szükséges `/var/state` hierarchiával; átmeneti intézkedésként az alkalmazás-specifikus adatok számára. Ám mindenképpen jegyezzük meg, hogy ez az engedmény a szabvány jövőbeni kiadásaiban megszűnik. Egy másik megoldás lehet, hogy a `/var/lib` egy szimbolikus link a `/var/state` könyvtárra.

BEGIN ÉRTELMEZÉS

A `/var/lib` könyvtár csak a object fájlok és ezek archíváltjai által használt fokozottan; ez igaz a jelenlegi BSD UNIX változatokra éppúgy, mint a jelenlegi GNU csomagokra. Ezek szerint a `/var/lib` elnevezés valóban nem tűnik megfelelőnek.

A BSD a `/var/db` elnevezést használja a hasonló könyvtára számára. Ez az elnevezés nyílt megszorításnak tűnik, mivel a bennefoglalt könyvtárstruktúrát elsődlegesen az adatbázis fájlok (`.db`) részére szánták.

END ÉRTELMEZÉS

5.11.1. /var/state/<szövegszerkesztő>: Szövegszerkesztői (editor) fájlmentések és állapot fájlok

Ezek a könyvtárak tartalmazzák azokat a mentett fájlokat, amelyeket egy szövegszerkesztő (például az elvis, jove, nvi) generált egy váratlan befejezésből kifolyólag.

Más szövegszerkesztők részére nem szükséges olyan könyvtár, amely az összeomlásból való visszaállításra szolgáló fájlok számára van fenntartva, de megkívánt egy jól definiált hely a további információk tárolására, amíg a szövegszerkesztő (*editor*) fut. Ezt az információt a `/var/state` alatti alkönyvtárban kellene tárolni (pédául a GNU Emacs elhelyezhetné a lezáró (*lock*) fájljait a `/var/state/emacs/lock` könyvtár alatt).

A jövőbeni szövegszerkesztők megkívánhatnak további állapot információt is; túl az összeomlásból való visszaállítást elősegítő fájlokon kívül, ezt az információt a `/var/state/<szövegszerkesztő>` könyvtár alatt kellene majd tárolni.

BEGIN ÉRTELMEZÉS

A Linux előző kiadásai – éppúgy mint a kereskedelmi eladók – használják a `/var/preserve` könyvtárat a vi és klónjai számára. Mindemelett valamennyi szövegszerkesztő a saját formátumát használja az összeomlásból való visszaállítást elősegítő fájljai számára, így valamennyi szövegszerkesztő részére külön alkönyvtár szükséges.

A szövegszerkesztő-specifikus lezáró (*lock*) fájlok általában eléggé eltérőek az eszközmeghajtó és az erőforrás lezáró fájloktól, amelyek a `/var/lock` könyvtárban tároltak, ezért ezek a `/var/state` alatt tároltak.

END ÉRTELMEZÉS

5.11.2. /var/state/misc: Különféle változó adatok

Ebbe a könyvtárba azok a változó adatok kerültek, amelyek nem lettek elhelyezve a `/var/state` valamelyik alkönyvtárában. Meg kellene kísérelni a relatívan egyedi elnevezéseket ebben a könyvtárban az elnevezésekből adódó konfliktusok elkerülésére.

Jegyezzük meg, hogy ennek a felépítésnek kellene tárolnia azokat az adatokat, amelyeket a jelenlegi BSD kiadások a `/var/db` könyvtárban tárolnak. Ezek közé értettek többek között a `locate.database` és a `mountdtab` fájlok, valamint a kernel szimbólum adatbázis(ok).

5.12. /var/tmp: Átmeneti fájlok, amelyek fenntartottak a rendszer bootolásai között

A `/var/tmp` könyvtár teszi elérhetővé a programok részére a szükséges átmeneti fájlok vagy könyvtárak fenntartását a rendszer bootolásai között. Emiatt a `/var/tmp` könyvtárban tárolt adatok álhatatosabbak fennmaradásuk szempontjából, mint a `/tmp` könyvtárban található adatok.

A `/var/tmp` könyvtár fájljait és könyvtárait nem szükséges törölni a rendszer bootolásakor. Habár a `/var/tmp` könyvtárban tárolt adatok általában gépre jellemző módon kerülnek törlésre, javallott, hogy az itt tárolt adatok kevésbé gyakori időközönként legyenek törölve, mint a `/tmp` könyvtár adatai.

5.13. /var/yp: A Hálózati Információs Szolgálat – NIS adatbázis fáljai

A Hálózati Információs Szolgálat (Network Information Service – NIS) változó adatai – amelyek régebben Sun Yellow Pages (YP) néven volt ismert – lettek elhelyezve ebben a könyvtárban.

BEGIN ÉRTELMEZÉS

A `/var/yp` a szabványos könyvtár a NIS (YP) adatai számára és majdnem kizárólagosan csak a NIS dokumentációban és rendszerekben használt.

A NIS nem tévesztendő össze a Sun NIS+-val, amely egy ettől eltérő `/var/nis` nevű könyvtárat használ.

END ÉRTELMEZÉS

6. Operációs rendszer specifikus melléklet

Ez a rész olyan további követelmények és ajánlások számára fentartott, amelyeket csak egy adott operációs rendszerre alkalmaznak. E fejezet anyagának sosem kellene ellentétben lennie az szabvány alapjával.

6.1. Linux

Ez a melléklet a Linux operációs rendszer részére készült.

6.1.1. /: A root könyvtár

A Linux rendszereken, amennyiben a kernel a gyökérben (/) van elhelyezve, javasoljuk a `vmlinux` vagy a `vmlinuz` nevek használatát, amelyek mai Linux kernelforrás csomagokban használtak.

6.1.2. /dev: Eszközmeghajtó és speciális fájlok

Valamennyi eszközmeghajtó és speciális fájl esetében `/dev` könyvtárban ragaszkodni kellene a Linuxra Megállapított Eszközmeghajtók (*Linux Allocated Devices*) dokumentumhoz, amely elérhető a Linux kernel forrásával együtt. E dokumentum fenntartója H. Peter Anvin <hpa@zytor.com>.

A `/dev` könyvtár szimbolikus hivatkozásai (*symbolic links*) ne legyenek terjesztve a Linux rendszerrel együtt, kivéve amelyeket a Linuxra Megállapított Eszközmeghajtók (LAD) dokumentum is tartalmaz.

BEGIN ÉRTELMEZÉS

Követelmény, hogy ne legyenek össze-vissza készített szimbolikus hivatkozások (*symlinks*), mivel a helyi beállítások gyakran el térnek majd attól az álapottól, amely a terjesztő fejlesztői gépén vannak. Bár előfordulhat, hogy egy terjesztés (*distribution*) installációs scriptje konfigurálja a szimbolikus hivatkozásokat, ezek a szimbólikus hivatkozások azonban gyakran nincsenek frissítve, ha helyi változtatások eszközöltek a hardverben. Mindemellet, amennyiben megbízhatóan használhatóak helyi szinten, valóban jobbá tudják tenni a használatot.

END ÉRTELMEZÉS

6.1.3. /proc: Kernel és folyamat (process) információs virtuális fájlrendszer

A proc fájlrendszer előnyösebb a szabványos Linux típusú folyamatok és a rendszerinformációk kezelésére, mint a `/dev/kmem` és más hasonló eljárások. Erősen javalljuk ezt a tároló és a folyamat információk visszakeresésére éppúgy, mint más kernel és memória információk esetében.

6.1.4. /sbin: Nélkülözhetetlen rendszerbinárisok

A Linux rendszerek ezen további fájlokat a `/sbin` könyvtárban helyezik el.

- Második kiterjesztésű fájlrendszer parancsok (opcionális):
  `{ badblocks, dumpe2fs, e2fsck, mke2fs, mklost+found, tune2fs }`
- Bootbetöltő (*boot-loader*) - térképfájl (

Az FHS levelezőlista Az FHS levelezőlista megtalálható az <fhs-discuss@ucsd.edu> címen. A listára való feliratkozáshoz egy „`ADD fhs-discuss`” levéltestű e-mailt kell küldeni a <listserv@ucsd.edu> címre.

Köszönet a San Diegoi University California Network Operations-nek, amely engedélyezte számunkra, hogy használjuk a kiváló levelező lista szerverét.

Amint az a bevezetőben is leírt, kérjük, ne küldj levelet a levelező listára anélkül, hogy előzőleg fel nem vetted volna a kapcsolatot az FHS szerkesztőjével vagy a felsorolt munkatársakkal.

Elismerések Az FHS fejlesztői köszönetet mondanak a fejlesztőknek, a rendszeradminisztrátoroknak és a felhasználóknak, akiknek információi nélkülözhetetlenek voltak e szabványhoz. Köszönetet kívánunk mondani valamennyi munkatársnak, akik segítettek írni, fordítani, és megalkotni ezt a szabványt.

Az FHS Csoport köszönetet kíván mondani azoknak a Linux fejlesztőknek, akik támogatták az FSSTND-t, e szabvány megelőzőjét. Ha ők nem demonstrálták volna az FSSTND előnyös voltát, az FHS-t valószínűleg sohasem lehetett volna kifejleszteni.

Közremüködők

Brandon S. Alley	<bsa@kf8nh.wariat.org>
Keith Bostic	<bostic@cs.berkeley.edu>
Drew Eckhardt	<drew@colorado.edu>
Rik Faith	<faith@cs.unc.edu>
Stephes Harris	<sweh@spuddy.mew.co.uk>
Ian Jackson	<ijackson@cus.cam.ac.uk>
John A. Martin	<jmartin@acm.org>
Ian McCloghrie	<ian@ucsd.edu>
Chris Metcalf	<metcalf@lcs.mit.edu>
Ian Murdock	<imurdock@debian.org>
David C. Niemi	<niemidc@clark.net>
Daniel Quinlan	<quinlan@pathname.com>
Eric S. Raymond	<esr@thyrsus.com>
Mike Sangrey	<mike@sojurn.lns.pa.us>
David H. Silber	<dhs@glowworm.firefly.com>
Theodore Ts'o	<tytso@athena.mit.edu>
Stephen Tweedie	<sct@dcs.ed.ac.uk>
Fred N. van Kempen	<waltje@infomagic.com>

Tartalomjegyzék