

Systemy plików LINUX

Przy instalacji Linuksa zwykle do wyboru jest kilka systemów plików, takich jak ext4, ext3, ext2, reiserfs, xfs, niekiedy może reiser4. Mandriva Linux w wersjach 10 i starszych oferowała do wyboru wszystkie obsługiwane przez nią systemy plików, na oko było to około 60 pozycji – w tym wszelkie rodzaje FAT i NTFS – przy czym instalator nie umożliwiał instalacji na większości z nich (nieśmiertelny komunikat "(/) musi być na porządnym systemie plików").

System plików – metoda przechowywania plików, zarządzania plikami, informacjami o tych plikach, tak by dostęp do plików i danych w nich zgromadzonych był łatwy dla użytkownika systemu.

Systemy plików stosuje się dla różnych nośników danych, takich jak dyski, dyskietki, a także w strumieniach danych, sieciach komputerowych, pamięciach.

We współczesnych systemach operacyjnych bezpośrednie operowanie na danych w plikach zarezerwowane jest tylko dla systemu operacyjnego, aplikacje mają zabroniony bezpośredni dostęp do nośnika danych.

Popularne systemy plików linuxa: Minix, xia, Ext, Ext2, Ext3, Ext4, umsdos, msdos, ReiserFS, vfat, XFS, proc, JFS, smb, NFS, ncp, Swap, iso9660, Sysv, hpfs, Affs, ufs Minix – pierwszy system plików wykorzystywany w linuxie, który został stworzony jeszcze przed powstaniem linuxa dla systemu operacyjnego o nazwie MINIX. Posiadał kilka ograniczeń:

- Długość nazwy pliku maksymalnie 14 znaków, (po pewnym czasie zwiększono do 30 znaków)
- Wielkość partycji nie większa niż 64 MB,

Ext (EXTENDED FILE SYSTEM) – następca Minix File System, dopuszczał pliki i partycje do rozmiaru 2 GB oraz długości nazw plików do 255 znaków. Miał jedną dużą wadę, która spowodowała, że nie stał się popularny: sposób zapamiętywania wolnych bloków i węzłów doprowadzał do znacznej fragmentacji dysku co wpływało negatywnie na wydajność systemu.

EXT2

Obecny w Linuksie od stycznia 1993, powstały na podstawie bardzo ograniczonego systemu plików Miniksa (początkowo ten system plików był używany na Linuksie); sam Linux był tworzony z użyciem tego systemu (nieśmiertelna [znowu] wiadomość Linusa anonsująca Linuksa). Nie będziemy się jednak o nim rozpisywać, ponieważ nie posiada on księgowania. Podobnie jak wszystkie wersje FAT.

W porównaniu do poprzednika ma **wiele zalet**:

- Obsługuje partycje o wielkości do 4 TB, pliki o wielkości do 2 GB,
- Rozpoznaje uszkodzenie systemu plików,
- Automatycznie naprawia uszkodzone sektory za pomocą programu e2fsck,

- Automatycznie sprawdza system po awarii i co określony czas, posiada wysoką **wydajność dzięki przeciwdziałaniu fragmentacji**

Wady:

- Długotrwałe sprawdzanie systemu plików po niepoprawnym zamknięciu,
- Niska wydajność dla bardzo małych plików,
- Mało efektywna obsługa katalogów.

EXT3

2

Obecny w Linuksie od listopada 2001, niektórzy nazywają go "aktualizacją EXT2". Główna zmiana to wprowadzenie księgowania. Wydajność stoi na dobrym poziomie, co można zobaczyć w podlinkowanym wyżej zestawieniu. Wprowadza kilka trybów księgowania, które pozwalają dostosować szybkość versus bezpieczeństwo do własnych zastosowań. Domyślnie działa w trybie ordered. O księgowaniu i zmianie trybu księgowania można poczytać w artykułach podlinkowanych wyżej.

Jak widać EXT3 posiada solidne podstawy (EXT2 w czasach swojej świetności był chętnie używany w zastępstwie filesystemu Miniksa) i jest już wypróbowany w działaniu. Co więcej, w razie potrzeby można wymusić jego zamontowanie jako EXT2 i użycie bez księgowania. Jest to system plików zdecydowanie godny polecenia dla początkującego użytkownika, nadaje się też na serwer.

Właściwości:

- Journaling - mechanizm księgowania zwiększający bezpieczeństwo systemu,
- Indeksowane katalogi - znacznie zwiększają wydajność systemu przy dużej ilości plików,
- Zapis synchroniczny - w najnowszych wersjach systemu Ext3 (jądro 2.4.19) działa ponad 10 razy szybciej od wersji z Ext2,
- Kompatybilność z Ext2

Obsługa ext3

Formatowanie ext3

Jeśli chcesz utworzyć partycję w formacie EXT3, musisz użyć narzędzia mkfs.ext3. Może jednak najpierw zobaczmy gdzie jest jakaś wolna partycja używając cfdisk:

Czasem trzeba cfdiskowi podać odpowiednie urządzenie np. cfdisk /dev/hdb (zobacz artykuł o montowaniu).¹ (*więcej informacji na stronie*).

Z wyjścia mkfs.ext3 oprócz ciekawych informacji (np. rozmiar klastra²: 4KiB, liczba zarezerwowanych bloków: 5%) dowiadujemy się o jeszcze jednym przydatnym narzędziu

¹ http://linux-porady.info/strony/dla_bardziej_zaaawansowanych/system_plikow_ext3_ordered_journal_ext4_reiserfs_notail_xfs_narzedzia: tune2fs fsck xfs fsr

² **Klastr** - jednostka logiczna wypełnienia dysku przez dane. Jeden klastr może zawierać tylko jeden plik, nawet jeśli plik jest mniejszy niż rozmiar klastra. Dlatego im mniejszy rozmiar klastra, tym większa oszczędność miejsca, ale zwykle mniejsza wydajność.

tune2fs. Zaraz się nim zajmiemy, ale najpierw zobacz, że wszystkie systemy plików tworzy się podobnie:

Większość poniższych opcji powinna działać także z EXT4.

Tune2fs pozwala nam - jak czytamy - zmienić domyślne ustawienia sprawdzania dysku. Mimo księgowania system plików ma zakodowane kompletne sprawdzenie poprawności danych, co jakiś czas, czyli przy 27 odmontowaniu lub upływie pół roku flaga clean ustawiana jest na unclean i system uruchamia sprawdzanie przy włączaniu. Więcej o tym niżej. Domyślne ustawienia są raczej dobre do serwerów, bo w domu przecież komputer wyłączany i włączany jest najczęściej kilka razy w ciągu dnia, więc owe 27 montowań powodowałoby sprawdzanie dysku średnio, co pół miesiąca.

Tune2fs zawiera też przełącznik -r, który w manualu opisany jest w taki sposób: Zmienia liczbę zarezerwowanych bloków na podanym urządzeniu. Na cóż one są zarezerwowane? Wyjaśnione jest to w manualu mkfs.ext3. Wolne miejsce na partycji pomaga utrzymać w ryzach fragmentację (jądro ma trochę wolnego miejsca na kompakowanie plików) a w razie zużycia całego miejsca przez jakiegoś użytkownika zarezerwowany obszar zabezpiecza możliwość dalszego logowania zdarzeń przez sysloga.

Tune2fs nadaje również etykiety. Spróbujmy go użyć do zmiany ustawienia sprawdzania systemu plików, na co 60 dni lub co 120 zamontowań w zależności od tego, co wystąpi pierwsze. Nadajmy partycji jakąś etykietkę. Zmieńmy wreszcie obszar zarezerwowany na 0 i porównajmy dostępne dla użytkownika miejsce na dysku:

Przyjrzyjmy się takiemu wpisowi w /etc/fstab:

`/dev/sda10 /home/ ext4 defaults,noatime,commit=100 1 2`

Zacznijmy może od tego, czemu system plików opisany został, jako ext4. Na partycji /home znajduje się system plików ext3, jednakże od jądra 2.6.27 (wersję można sprawdzić poleceniem `uname -a`) możliwe jest wykorzystanie modułu ext4 od obsługi ext3. Dzięki temu przyspieszysz alokację większych plików oraz zmniejszysz fragmentację dzięki opóźnionej alokacji (działa lepiej w połączeniu z większym commit interval). Użycie modułu ext4 na systemie plików ext3 nie spowoduje zmiany formatu na ext4, takiej partycji można później swobodnie używać na systemie nieobsługującym ext4. Opcja noatime spowoduje, że fakt dostępu do każdego pliku nie będzie zapisywany w systemie plików. Powinno to spowodować redukcję liczby zapisów na dysk i przyspieszenie otwierania katalogów z wieloma elementami.

Commit interval wyrażany w sekundach to czas, po jakim dane z buforów dyskowych zostaną zapisane fizycznie na dysk. Domyślnie jest to 5 sekund. Zwiększenie tego czasu w znaczny sposób podnosi wydajność, jednak przy awarii zasilania spowoduje utratę danych zbuforowanych między kolejnymi cyklami zapisu.

Do sprawdzania poprawności systemu plików³ służy program `fsck`, `fsck.ext3` lub `e2fsck`. Właściwie wychodzi na jedno, czego użyjemy, i tak uruchomiony zostanie w końcu `e2fsck`. Czemu tak śmiesznie? Zaczniemy od końca. `E2fsck` to po prostu właściwa nazwa programu do sprawdzania poprawności `ext2`, `ext3` i `ext4`. `Fsck.ext3` to ukłon w stronę użytkownika, który musi znać tylko polecenie `fsck`:

Fsck jest używany w skryptach startowych, gdzie rozpoznaje typ systemu plików i uruchamia odpowiednie narzędzie. Przed użyciem `e2fsck` trzeba koniecznie odmontować partycję. Aby przekonać się o tym samym efekcie końcowym, spróbujmy sprawdzić dysk na wymienione 3 sposoby:

```
blueice root [/home/mk] # fsck /dev/sda5
fsck 1.40.1 (08-Jul-2007)
e2fsck 1.40.1 (08-Jul-2007)
pornole: clean, 11/788704 files, 60641/1574354 blocks
blueice root [/home/mk] # fsck.ext3 /dev/sda5
e2fsck 1.40.1 (08-Jul-2007)
pornole: clean, 11/788704 files, 60641/1574354 blocks
blueice root [/home/mk] # e2fsck /dev/sda5
e2fsck 1.40.1 (08-Jul-2007)
pornole: clean, 11/788704 files, 60641/1574354 blocks
```

i dla zobrazowania, że `e2fsck` służy też do sprawdzania `ext2`:

```
blueice root [/home/mk] # fsck.ext2 /dev/sda5
e2fsck 1.40.1 (08-Jul-2007)
pornole: clean, 11/788704 files, 60641/1574354 blocks
```

Tak właściwie `e2fsck` sprawdził tylko, że system plików oznaczony jest flagą `clean` i odmówił w związku z tym dalszego sprawdzania. Wyżej w akapicie o konfiguracji `tune2fs` było powiedziane o tym, że ta flaga zostaje ustalona na `unclean` po 27 odmontowaniu (my ustaliliśmy tę wartość na 120) i przy uruchomieniu systemu następuje wymuszenie sprawdzenia konsystencji mimo księgowania. Zobaczmy jak to działa:

```
blueice root [/home/mk] # fsck /dev/sda5
fsck 1.40.1 (08-Jul-2007)
e2fsck 1.40.1 (08-Jul-2007)
pornole has been mounted 120 times without being checked, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
pornole: 11/788704 files (9.1% non-contiguous), 60641/1574354 blocks
```

³ **Manuale** - większość przedstawionych tutaj poleceń posiada przystępną pomoc po polsku dostępną z konsoli po wpisaniu `man program` lub w pasku adresu Konquerora `man:program`.

Po tym sprawdzeniu flaga unclean zostanie znowu zmieniona na clean i partycja przy starcie nie będzie sprawdzana. Sprawdzenie można także wymusić na żądanie (tym razem chcemy wiedzieć także więcej informacji - opcja -v).

```
blueice root {/home/mk} # fsck.ext2 /dev/sda5 -f -v
e2fsck 1.40.1 (08-Jul-2007)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
11 inodes used (0.00%)
1 non-contiguous inode (9.1%)
# of inodes with ind/dind/tind blocks: 0/0/0
60641 blocks used (3.85%)
0 bad blocks 1 large file
0 regular files
2 directories
0 character device files
0 block device files
0 fifos
0 links
0 symbolic links (0 fast symbolic links)
0 sockets ----- 2 files
```

ReiserFS

Stworzony specjalnie dla Linuksa przez firmę Namesys w 2001 roku zwany także Reiser3 to system plików zaprojektowany i zaimplementowany przez grupę kierowaną przez Hansa Reisera. ReiserFS jest obecnie obsługiwany przez GNU/Linuksa i może być w przyszłości włączony do innych systemów operacyjnych. ReiserFS to jeden z pierwszych systemów plików z księgowaniem dla GNU/Linuksa. używa mechanizmów podobnych do bazy danych, co powoduje szybszy dostęp do dużej ilości małych plików. Księgowanie podobne jest w końcowym efekcie do ext3 ordered. Dodatkowo pozwala na użycie pojedynczego klastra do przechowywania kilku plików, co efektywnie zwiększa ilość miejsca dostępnego dla użytkownika i zmniejsza wewnętrzną fragmentację⁴. Wyłączenie tej opcji i zwiększenie szybkości uzyskujemy opcją notail.

Dzięki wspomnianej funkcji efektywnie zmniejsza się również wartość fragmentacji danych (czyli tej, którą usuwają defragmentatory). Reiserfs to najmniej fragmentujący się popularny system plików dla Linuksa wśród uznanych za stabilne.

⁴ **Wewnętrzna fragmentacja** - zarezerwowanie przestrzeni bez intencji jej użycia. Oprócz wspomnianej sytuacji z klastrami podobnie ma się z programami, które rezerwują dla zmiennych zwykle więcej pamięci RAM, niż to jest im faktycznie potrzebne. Powodem tego zjawiska jest chęć uproszczenia budowy rozwiązań informatycznych.

W porównaniu podlinkowanym wyżej można sprawdzić dokładnie gdzie reiserfs jest lepszy, a gdzie gorszy od ext3. Jak widać teoretyczne założenia nie zawsze idą w parze z praktyką. Dodatkowo reiserfs zużywa więcej czasu procesora do działania. Warto też wspomnieć, że główny developer, założyciel Namesys, Hans Reiser siedzi w więzieniu za zabójstwo żony, więc o ile reiserfs znajduje się już w jądrze Linuksa, o tyle ma na pewno mniejsze wsparcie niż gdyby jego autor był na wolności.

Reiserfs jest również mniej odporny na uszkodzenia struktury. W razie niepowodzenia jej naprawy z kroniki zwykle przebudowane musi zostać całe drzewo systemu plików co nie zawsze się udaje i może być skutecznie zakłócone istniejącym na przebudowywanej partycji obrazem jakiejś partycji z reiserfs.

Nie twierdzę, że reiserfs jest zły, ale wydaje mi się, że ext3 nie będzie gorszy w większości zastosowań, a jego prosta budowa i większa dojrzałość zapewniają lepsze bezpieczeństwo danych. Z tego powodu i ponieważ ogólny wzór obsługi systemu plików został podany w dziale o EXT3, ograniczę się tu jedynie do napisania, że utworzyć system plików reiserfs można za pomocą polecenia `mkfs.reiserfs`, sprawdzić przez `reiserfsck`, a skonfigurować przy użyciu `reiserfstune`. Opcję `notail` włącza się przy montowaniu.

Nie będziemy zajmować się też reiserem4, który nie został włączony do jądra Linuksa, choć firma Namesys uznała go za stabilny. Jego fani zapewne znajdą sposób na jego włączenie w jądro i przetestowanie. Nikt rozsądny jednak nie używa go na maszynie produkcyjnej.

ReiserFS - Właściwości:

- Bardzo efektywny sposób przechowywania wszystkich informacji o plikach i katalogach,
- Zaawansowany system transakcji zapewniający spójność wszystkich danych zapisanych w systemie,
- Kompresja wielu małych plików oraz tzw. ogonów (końcówek plików o rozmiarze mniejszym od wielkości bloku) w jednym bloku dyskowym pozwalająca w znacznym stopniu zminimalizować fragmentację wewnętrzną,
- Efektywna obsługa nawet dużych katalogów (stosowana jest w tym przypadku tablica haszująca, dla której klucze są generowane na podstawie nazwy pliku - bardzo szybkie wyszukiwanie pliku),
- Zaawansowany system wtyczek (plugins) pozwalający niemal dowolnie modyfikować zachowanie warstwy semantycznej systemu (w przeszłości także w pewnym stopniu fizycznej),

Reiser4 - wersja tworzona od podstaw, w porównaniu do reiserFS zawiera:

- Zmiane architektury systemu na bardziej obiektowy,
- Używanie repackera - specjalnego programu, który upakuje ogony, jeszcze bardziej oszczędzając miejsce,

- Zwiększoną wydajność,
- Lepsze zapewnienie bezpieczeństwa,
- Możliwość dołączania własnych pluginów.

Reiser4 wprowadza innowację w zakresie bezpieczeństwa – dzieli plik na mniejsze części, z których każda może mieć indywidualne uprawnienia i korzystać z różnych dodatków. Wadą Reiser4 jest wolne usuwanie danych, jednak deklasuje on rywali, jeśli chodzi o zapisywanie wielu strumieni danych w jednym czasie. Nadają się także idealnie do przechowywania dużej ilości małych plików, oszczędzając przy tym dużo miejsca.

XFS

Wyprodukowany w 1994 roku przez firmę Silicon Graphics dla IRIX-a. Zwycięża szybkością w prawie każdym zestawieniu, jednak jego księgowanie w przypadku awarii zasilania nie zapewnia bezpieczeństwa danych, a jedynie sprawność struktury systemu plików. Zawiera wiele ciekawych mechanizmów takich jak określenie minimalnego transferu dla konkretnego pliku, rozszerzone uprawnienia i inne nieprzydatne w domu funkcje. Do rozważenia w przypadku mało istotnych rzadko zapisywanych danych lub jeśli jest się szczęśliwym posiadaczem UPS-a, czy laptopa.

Obsługa XFS-a

XFS w odróżnieniu od innych systemów plików sprawdzany jest przy montowaniu, tak, więc polecenie **fsck.xfs** nic nie robi. Aby sprawdzić i naprawić XFS-a używane są polecenia `xfs_check` i `xfs_repair`. Na prawidłowym systemie plików, który był wcześniej zamontowany `xfs_check` nie powinien niczego wyświetlić. W przeciwnym razie system plików może być naprawiony od razu przy użyciu `xfs_repair` lub bezpieczniej za pomocą `xfsdump` i `mkfs.xfs`: najpierw zrzucaamy obraz partycji do pliku, tworzymy na jej miejsce nowy system plików, a później używamy `xfsrestore`, aby przywrócić dane na nowy system plików. `xfsdump` nie zadziała jednak na uszkodzonym systemie plików. Jeśli system plików jest montowalny, `xfsdump` może zostać użyty do odzyskania cennych plików przed zastosowaniem `xfs_repair`. W przypadku, gdy partycji nie można zamontować, `xfs_repair` to jedyne rozwiązanie.

XFS posiada wiele opcji konfiguracyjnych i funkcji, jednak inaczej niż w przypadku ext3, zwykły użytkownik nie ma raczej powodu do ich ruszania. Dla porządku podaję jednak dostępne możliwości:

`blueice root {/home/mk} # xfs [WCISŃIJ TAB 2 razy]`

<code>xfs xfs_copy</code>	<code>xfs_fsr</code>	<code>xfs_logprint</code>	<code>xfs_repair</code>	<code>xfsinvutil</code>
<code>xfs_admin</code>	<code>xfs_db</code>	<code>xfs_growfs</code>	<code>xfs_mkfile</code>	<code>xfs_rtcp</code>
<code>xfsrestore</code>	<code>xfs_bmap</code>	<code>xfs_estimate</code>	<code>xfs_info</code>	<code>xfs_ncheck</code>
<code>xfsdump</code>	<code>xfs_check</code>	<code>xfs_freeze</code>	<code>xfs_io</code> <code>xfs_quota</code>	<code>xfsinfo</code>

EXT4

System plików powstały na bazie EXT3, kompatybilny z nim (partycję ext3 można zamontować jako ext4 i na odwrót, pod warunkiem że ext4 nie używa ekstentów, czyli upychania kilku plików na jeden klaster jak w reiserfs). 25 grudnia 2008 został wydany jako stabilny wraz z jądrem 2.6.28, mimo wszystko zalecam ostrożność w jego używaniu.

Z ciekawych funkcji ext4 można wymienić możliwość defragmentacji bez odmontowywania, szybsze sprawdzanie poprawności, jeszcze lepsze bezpieczeństwo (przynajmniej w teorii dzięki sumom kontrolnym kroniki) i większą szybkość (połączenie writeback z sumami kontrolnymi. Trzeba podkreślić, że migracja EXT3 > EXT4 jest możliwa i nie powoduje utraty wydajności w porównaniu z "czystym" ext4, natomiast kosztuje utratę kompatybilności z zewnętrznymi narzędziami obsługującymi ext3 (na Windows na przykład). Powrót EXT 4 > EXT 3 nie jest możliwy konwencjonalnymi sposobami.

SWAP

To partycja wymiany, tworzy ona pamięć wirtualną, która jest rozszerzeniem pamięci RAM. Dzięki użyciu "partycji wymiany" RAM zwiększa swoją wydajność a co za tym idzie wydajność systemu również wzrasta. Zamiast partycji SWAP, system może utworzyć plik wymiany, jednak wymiana danych jest o wiele wolniejsza i wydajność systemu drastycznie spada.

NFS

To usługa pozwalająca udostępniać zasoby dyskowe komputerom w sieci. Serwer udostępnia katalogi klientom, którzy mogą je podmontować i działać jak na lokalnym systemie plików. Montowanie: `mount serwer.net:/usr/local /usr/local -t nfs`. JFS (Journaled Filesystem)

Właściwości:

- Obsługa bardzo dużych plików i partycji (rzędu kilku tysięcy TB),
- Dynamiczna alokacja i-węzłów,
- Efektywne wyszukiwanie i-węzłów, wolnego miejsca na dysku oraz plików w katalogach,
- Zaawansowany system kronikowania operacji dyskowych,
- Wbudowany bezpośrednio w system plików, a nie dodany do niego później jak w przypadku innych systemów plików,
- Zastosowanie przedziałów bloków dyskowych (Extent) w celu zmniejszenia fragmentacji danych i zwiększenia efektywności odwołań do nich,
- Przechowywanie małych plików (linków symbolicznych) oraz katalogów bezpośrednio w i-węzłach co znacznie zwiększa efektywność odwołań do nich,
- Efektywna obsługa rozrzedzonych plików.

I-węzeł (inode)

To struktura danych opisująca pliki zapisane na dysku. I-węzeł zawiera informacje o: typie pliku, właścicielu, grupie, długości pliku, adresie, liczbie dowiązań i kilka innych informacji. Długość pliku to jego wielkość, ilość zajmowanego miejsca na dysku.

Adres to numer sektora, w którym znajduje się plik. Adresem są sektory dysku, w których należy szukać segmentów pliku (zwykle te sektory mają wielkość 512 bajtów). Liczba dowiązań jest to ilość skojarzonych "kopii" pliku ze zbiorem danych. XFS - 64-bitowy system plików zaprojektowany przez firmę Silicon Graphics Inc. z przeznaczeniem do użycia go w systemie operacyjnym IRIX (wersja UNIX-a firmy SGI). Aktualnie jest dostępna również jego implementacja dla systemu Linux rozwijana przez SGI jako projekt na licencji wolnego oprogramowania.

Właściwości:

- pozwala na obsługę dużych dysków twardych, (Maksymalny rozmiar woluminu jest ograniczony do 18 milionów TB.),
- rozmiar pojedynczego pliku może wynosić maksymalnie 2^{63} bajtów czyli ponad 8 milionów TB (dokładnie 8 388 608 TB).
- rozmiar jednostki alokacji może wynosić od 512 bajtów (wielkość fizycznej jednostki alokacji) do 1 MB.

System plików, partycje i dyski

Organizacja systemu plików pod Linuksem nieco różni się od tego co znamy z systemów MS Windows. W tym artykule postaramy się uwypuklić co najważniejsze różnice, tak aby hierarchia linuksowego systemu plików jak i sposób zarządzania partycjami i dyskami nie sprawiał Wam kłopotu.

Podstawy systemu plików, – czyli gdzie jest ten Program Files??

Po pierwsze nie mamy w drzewie katalogów wirtualnych zasobów, jak np. w systemie Windows dyski C, D, dyskietka, cdrom, etc. Wszystkie zasoby w drzewie katalogów Linuksa zaczynają się od magicznego /, co nazywamy katalogiem root, czyli katalogiem korzeniem (nie mylić z użytkownikiem root – to zupełnie inne sprawy).

Tak więc, jeśli mamy w systemie kilka partycji, możemy zamontować je jako dowolny katalog w drzewie. Np. jeśli całą partycję chcemy przeznaczyć na katalog domowy, to montujemy ją w punkcie /home. I tak dalej. Podobnie z urządzeniami zewnętrznymi, jak dyski CD, dyskietki czy pamięci USB. Według standardu FHS, powinny być one zamontowane w katalogu /media. Np. /media/cdrom0, /media/fd0, etc. W starszych dystrybucjach, często urządzenia te montowane są bezpośrednio w root, np. /cdrom, /floppy. Często też tworzone są linki symboliczne (skrót) w obu miejscach.

Inna jest również filozofia jeśli chodzi o lokalizację instalowanych w systemie programów. Ze względu na modułowość systemu (aplikacje korzystają zazwyczaj z wielu innych aplikacji – pakietów), programy nie instalują się w odseparowanych katalogach (a’la windowsowe Program Files). Zamiast tego, odpowiednie pliki umieszczane są w odpowiednich miejscach w systemie, tak aby mogły być użyte przez inne aplikacje (głównie chodzi tu o użyte w pakiecie biblioteki, ale również o pliki binarne). Zamiast Program Files mamy więc katalogi bin na pliki wykonywalne, lib na biblioteki, doc na dokumentację, i wiele innych, odpowiednio do zastosowania.

Tyle teorii. W poniższej tabelce natomiast lista linuksowych odpowiedników najpopularniejszych katalogów systemowych systemu Windows.

10

	MS Windows	GNU/Linux
Root	C:\	/
Katalog domowy	C:\Documents and settings\USER\Moje dokumenty	/home/USER (np. /home/zdzisiek)
Pliki konfiguracyjne	Rejestry, dostępne przez polecenie regedit	/etc
Pliki systemowe	C:\Windows	Pliki wykonywalne: /bin, /sbin Biblioteki systemowe: /lib
Zainstalowane aplikacje	C:\Program Files	Pliki wykonywalne: /usr/bin, /usr/local/bin, etc Biblioteki: /usr/lib, /usr/local/lib, /usr/share/lib, etc Dokumentacja: /usr/share/doc/, /usr/doc, etc
CDROM	D:\ (lub kolejny wirtualny dysk)	/media/cdrom (alternatywnie /mnt/cdrom lub /cdrom)
Dyskietka	A:\	/media/fd0 (alternatywnie /mnt/fd0 lub /floppy)

Montowanie dysków

Wszystkie urządzenia w systemie muszą być zamontowane przed użyciem i wymontowane przed odłączeniem. System Windows robi to zazwyczaj automatycznie. Linux też może. Domyślnie jednak takie zasoby jak CDROM-y, dyskietki czy napędy USB nie są (z wyjątkiem kilku dystrybucji) montowane automatycznie (nawet po włożeniu do zasobu do napędu). W takim przypadku, musimy ręcznie wydać polecenie `mount` wraz z nazwą katalogu montowania zasobu (zdefiniowanego w `/etc/fstab`), np.

`mount /media/cdrom`

zamontuje nam (w naszej przykładowej konfiguracji) CDROM-a. Podobnie

`umount /media/cdrom`

Odmontuje zasób – od tego momentu będziemy mogli wyjąć płytkę CD z napędu.

Jak zamontować partycje Windows?

Windows wspiera dwa systemy plików. FAT – starszy i bardziej prymitywny – wspierany jest pod Linuksem (możemy odczytywać i zapisywać dane na partycjach Windows z tym systemem plików bez większego zachodu). Z NTFS długo było nieco gorzej, ale w 2007 roku poprawiło się to znacznie. Mimo to, w większości dystrybucji domyślnie wspierany jest tylko odczyt z takiej partycji. Do zapisu będziemy potrzebować oddzielnego programu — `ntfs-3g`. Jego instalacja w Ubuntu jest opisana w naszym FAQ: Instalacja obsługi zapisu na partycjach NTFS. Aby zamontować partycję Windows musimy najpierw wiedzieć, w jaki sposób system Linux ją widzi. Możemy do tego użyć programu `fdisk`, który służy do zarządzania dyskami partycjami. Przykładowy output komendy⁵:

```
# fdisk -l /dev/hda

Disk /dev/hda: 60.0 GB, 60011642880 bytes
255 heads, 63 sectors/track, 7296 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot   Start      End   Blocks  Id System
/dev/hda1             1      1211    9727326   7  \
HPFS/NTFS
/dev/hda2          1216      7296   48845632+   5  \
Extended
/dev/hda5    *    1216      3040   14648224+  83  \
Linux
/dev/hda6          3041      3679    5132736   b  \
W95 FAT32
/dev/hda7          3680      4049    2971993+  83  \
Linux
/dev/hda8          4050      4319    2168743+  82  \
Linux swap / Solaris
/dev/hda9          4320      7296    23908216+  83  \
Linux
```

W tym przypadku, mamy dostępne 2 partycje Windows, pierwsza widoczna jako `hda1` (NTFS), a druga to `hda6` (FAT32). Aby je zamontować wystarczy wydać polecenia:

⁵ <http://jakilinux.org/pierwsze-kroki/system-plikow-partycje-i-dyski/#>

```
# mount -t ntfs /dev/hda1 /mnt/C
# mount -t vfat /dev/hda6 /mnt/D
```

Takie montowane partycje Windows ma zasadniczo dwie wady: trzeba je za każdym razem montować ręcznie oraz dostęp do nich ma tylko użytkownik root.

Możemy więc pójść o krok dalej i dopisać obie partycje do pliku `/etc/fstab`, tak aby ich montowaniem zajmował się system. Odpowiednie wpisy w `/etc/fstab` będą wyglądały mniej więcej tak:

```
/dev/hda1      /mnt/hda1      ntfs      uid=1000,gid=1000  \\
0              2
/dev/hda6      /mnt/hda6      vfat      \\
codepage=852,uid=1000,gid=1000,icharset=iso8859-2  \\
0              2
```

12

Opcje uid i gid oznaczają ID użytkownika i grupy, dla których partycje będą dostępne. Dla partycji FAT32 musimy podać dodatkowo domyślne kodowanie, tak aby polskie znaki diakrytyczne wyświetlały się bez problemów.

Jak zamontować pendrive'a?

W niektórych dystrybucjach (jak np. Ubuntu) obsługa pendrive i innych pamięci przenośnych (FLASH) skonfigurowana jest w ten sposób, że urządzenie automatycznie pojawia się jako ikonka na pulpicie tuż po włożeniu go do portu USB.

Zasoby, USB dostępne są zazwyczaj, jako wirtualne urządzenia `/dev/sda1` lub `/dev/sdb1`.

Wystarczy, więc w konsoli wpisać jako root:

```
mount -t vfat /dev/sda1 /mnt/pendrive
```

Pendrive zamontuje nam się w katalogu `/mnt/pendrive` (o ile taki katalog wcześniej stworzyliśmy). Proces możemy zautomatyzować przez dodanie odpowiedniej linii do pliku `/etc/fstab`. Będzie to wyglądać mniej więcej tak:

```
/dev/sda1 /mnt/pendrive vfat uid=1000,gid=1000,\\
exec,rw,codepage=852,icharset=iso8859-2 0 0
```

Teraz, aby zamontować dysk przenośny wystarczy wpisać **mount /mnt/pendrive** lub kliknąć na dysk w menadżerze plików Gnome lub KDE. Ustawienie kodowania jest potrzebne, jeśli mamy na penie pliki z polskimi znakami. Oczywiście, jeśli używamy na pendrive innego systemu plików niż FAT, trzeba to również wziąć pod uwagę przy montowaniu dysku.

Więcej o systemie plików i montowaniu urządzeń⁶

- [Montowanie windowsowych partycji \(FAT/NTFS\) w Linuksie](#) (ze strony DUG)
- [Dokładny opis systemu katalogów systemowych Linuksa](#) z Gery.pl
- [Urządzenia USB w Linuksie bez tajemnic \[PDF, 85 kB\]](#) - artykuł z Linux Magazine

⁶ <http://jakilinux.org/pierwsze-kroki/system-plikow-partycje-i-dyski/>