[ARCHIWUM] Atrybuty plików i katalogów

> **Strona zarchiwizowana.** Treść została wchłonięta i rozbudowana. Aktualna wersja: Atrybuty rozszerzone: chattr, lsattr. Ta strona zostaje wyłącznie dla historii.

Wprowadzenie:

Poza standardowymi uprawnieniami (rwx), system Linux oferuje dodatkowe mechanizmy kontroli dostępu do plików i katalogów. Należą do nich pliki ukryte oraz rozszerzone atrybuty zarządzane poleceniem `chattr`.

1. Pliki ukryte w systemie Linux/Unix

Co to są pliki ukryte?

W systemach Unix/Linux plikami ukrytymi nazywamy pliki i katalogi, których nazwa zaczyna się od kropki (`.`). Są one domyślnie niewidoczne przy standardowym wywoływaniu `ls`. Pliki ukryte często zawierają konfiguracje użytkownika lub aplikacji.

Wyświetlanie plików ukrytych:

```
ls -a
```

Pokazuje wszystkie pliki, w tym ukryte.

```
ls -la
```

Szczegółowy widok ze wszystkimi plikami, w tym ukrytymi.

Przykłady typowych plików ukrytych:

- `.bashrc` – konfiguracja powłoki bash
- `.bash_profile` – konfiguracja przy logowaniu
- `.ssh/` – katalog z kluczami SSH
- `.gitignore` – plik konfiguracyjny git
- `.vimrc` – konfiguracja edytora vim

Tworzenie ukrytego pliku:

```
touch .ukryty_plik.txt
```

Wystarczy, że nazwa zaczyna się od kropki.

2. Rozszerzone atrybuty plików – `chattr`

Co to są rozszerzone atrybuty?

Rozszerzone atrybuty to dodatkowe flagi kontrolujące zachowanie plików na poziomie systemu plików. Pozwalają one na bardziej precyzyjną kontrolę niż standardowe uprawnienia rwx. Są obsługiwane głównie w systemach plików ext2, ext3, ext4.

Komenda `chattr` – zmiana atrybutów:

```
chattr +i plik.txt
```

Ustawia atrybut niezmienności (immutable). Plik nie może być modyfikowany, usuwany, zmieniany ani przenoszony – nawet przez użytkownika root!

```
chattr -i plik.txt
```

Usuwa atrybut niezmienności.

3. Najważniejsze atrybuty `chattr`

Atrybut `i` (immutable – niezmienny):

```
chattr +i /etc/ważny_config.conf
```

Plik nie może być w żaden sposób modyfikowany. Przydatne do ochrony krytycznych plików konfiguracyjnych przed przypadkową zmianą.

Atrybut `a` (append only – tylko dopisywanie):

```
chattr +a /var/log/aplikacja.log
```

Plik może być tylko rozszerzany (dopisywanie na końcu), ale nie można modyfikować ani usuwać istniejącej zawartości. Idealne dla plików logów.

Atrybut `c` (compressed – kompresowany):

```
chattr +c duży_plik.txt
```

Jądro automatycznie kompresuje plik na dysku (jeśli system plików to obsługuje).

Atrybut `d` (no dump – pomijany w backup):

```
chattr +d cache.tmp
```

Plik jest pomijany przez narzędzie `dump` podczas tworzenia kopii zapasowej.

Atrybut `s` (secure deletion – bezpieczne usuwanie):

```
chattr +s tajny_dokument.txt
```

Przy usuwaniu pliku, jego bloki na dysku są nadpisywane zerami, co utrudnia odzyskanie danych.

Atrybut `u` (undeletable – możliwy do odzyskania):

```
chattr +u ważny_plik.txt
```

System zachowuje informacje pozwalające na odzyskanie pliku po usunięciu (jeśli system plików to wspiera).

4. Komenda `lsattr` – wyświetlanie atrybutów

```
lsattr plik.txt
```

Pokazuje aktualne atrybuty pliku.

```
lsattr -d katalog/
```

Pokazuje atrybuty samego katalogu (a nie jego zawartości).

```
lsattr -a
```

Pokazuje atrybuty wszystkich plików, w tym ukrytych.

Przykładowy wynik:

```
$ lsattr plik.txt
----i---------e----- plik.txt
```

Litera `i` oznacza atrybut immutable.

5. Rekurencyjna zmiana atrybutów

```
chattr -R +i /etc/config/
```

Opcja `-R` (recursive) stosuje atrybut do wszystkich plików i podkatalogów.

6. Praktyczne przykłady zastosowań

```
# Ochrona pliku konfiguracyjnego przed modyfikacją
sudo chattr +i /etc/ssh/sshd_config
# Teraz nawet root nie może go zmienić bez usunięcia atrybutu

# Sprawdzenie atrybutów
lsattr /etc/ssh/sshd_config

# Aby móc edytować, trzeba najpierw usunąć atrybut
sudo chattr -i /etc/ssh/sshd_config
```

```
# Zabezpieczenie pliku logów – tylko dopisywanie
sudo chattr +a /var/log/aplikacja.log

# Ochrona całego katalogu przed usunięciem
sudo chattr -R +i /opt/produkcja/

# Weryfikacja atrybutów wszystkich plików w katalogu
lsattr -a /home/user/
```

7. Ważne uwagi

- Rozszerzone atrybuty działają na poziomie systemu plików – nie wszystkie systemy plików je obsługują
- Tylko użytkownik root może zmieniać większość atrybutów
- Atrybuty są niezależne od standardowych uprawnień Unix (rwx)
- Podczas przenoszenia plików między systemami atrybuty mogą zostać utracone
- Warto dokumentować użycie atrybutów, aby inni administratorzy wiedzieli o ich zastosowaniu

---

Revision #3
Created 2025-11-23 14:20:48 UTC by Tomasz Konieczny
Updated 2026-09-17 21:48:13 UTC by Tomasz Konieczny