ReiserFS i Reiser4

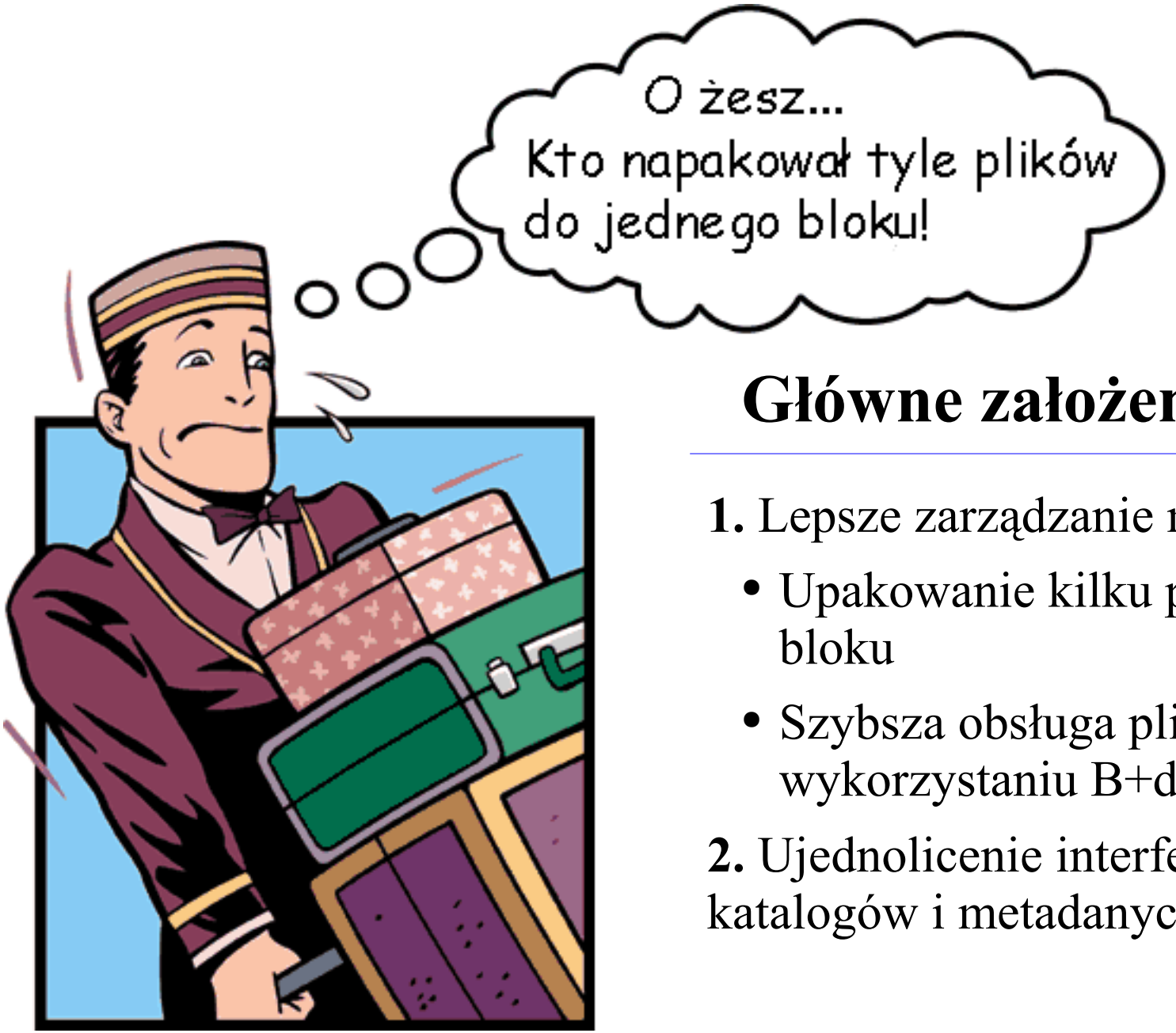
Dominik Klimczak

Wyzwania stojące przed systemami plików

- Obsługa coraz większych dysków i plików
- Obsługa długich nazw plików i ścieżek dostępu
- Prawa dostępu i możliwość przechowywania dodatkowych metainformacji
- Szybkość (efektywność stosowanych struktur danych)
- Gospodarność (minimalizacja rozmiaru metadanych i strat przy rezerwacji blokami)
- Radzenie sobie z fragmentacją

Wyzwania c.d.

- Odporność na awarie:
  - Transakcje (operacje na systemie plików mogą być naniesione tylko w całości lub w ogóle; są atomowe)
  - Kronikowanie (zapisywanie dziennika operacji zleconych systemowi, umożliwia podniesie się systemu po awarii) – możliwe kronikowanie tylko metadanych lub danych z metadanymi
- Kompresja plików “w locie”
- Obsługa szyfrowania



Główne założenia Reisera

1. Lepsze zarządzanie małymi plikami

- Upakowanie kilku plików w jednym bloku
- Szybsza obsługa plików dzięki wykorzystaniu B+drzew

2. Ujednolicenie interfejsu dla plików, katalogów i metadanych

ReiserFS

- Stworzony w 1996 roku przez grupę pod kierownictwem Hansa Reisera
- Licencja GPL
- Domyślny system plików dla SuSE, Gentoo
- Powstał z myślą o Linuksie
- Istnieje implementacja pod Windows: *http://rfsd.sourceforge.net/*
- Rozważany jako domyślny system dla następców BeOSa

Cechy ReiserFS

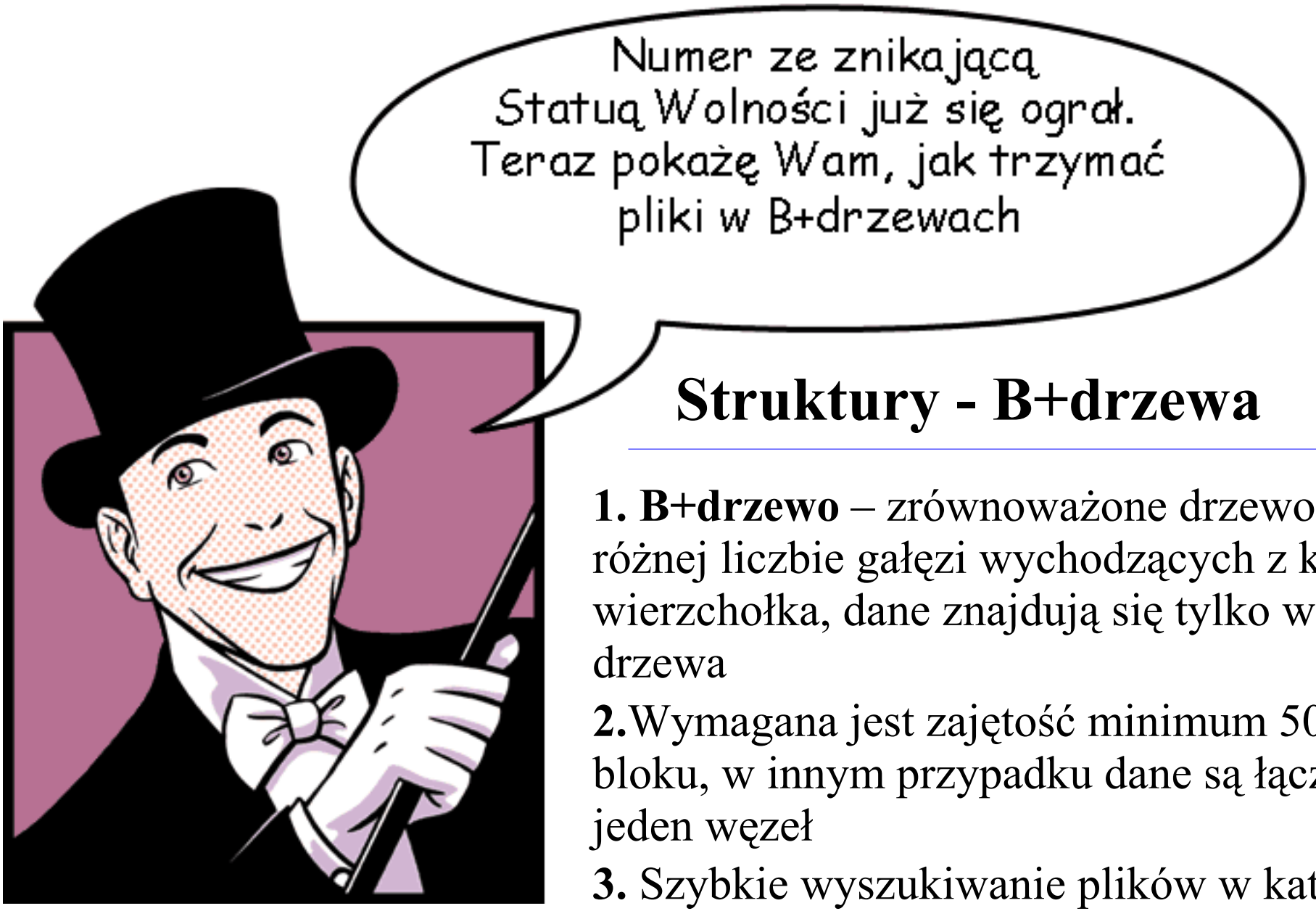
1. Kronikowanie (domyślnie tylko metadanych)

2. Możliwość rozszerzania już istniejących partycji

3. Atomowość operacji na systemie plików

4. Wydajność operacji na dużej ilości małych plików (zwłaszcza w jednym katalogu)

5. Możliwość redukcji fragmentacji wewnętrznej plików



Struktury - B+drzewa

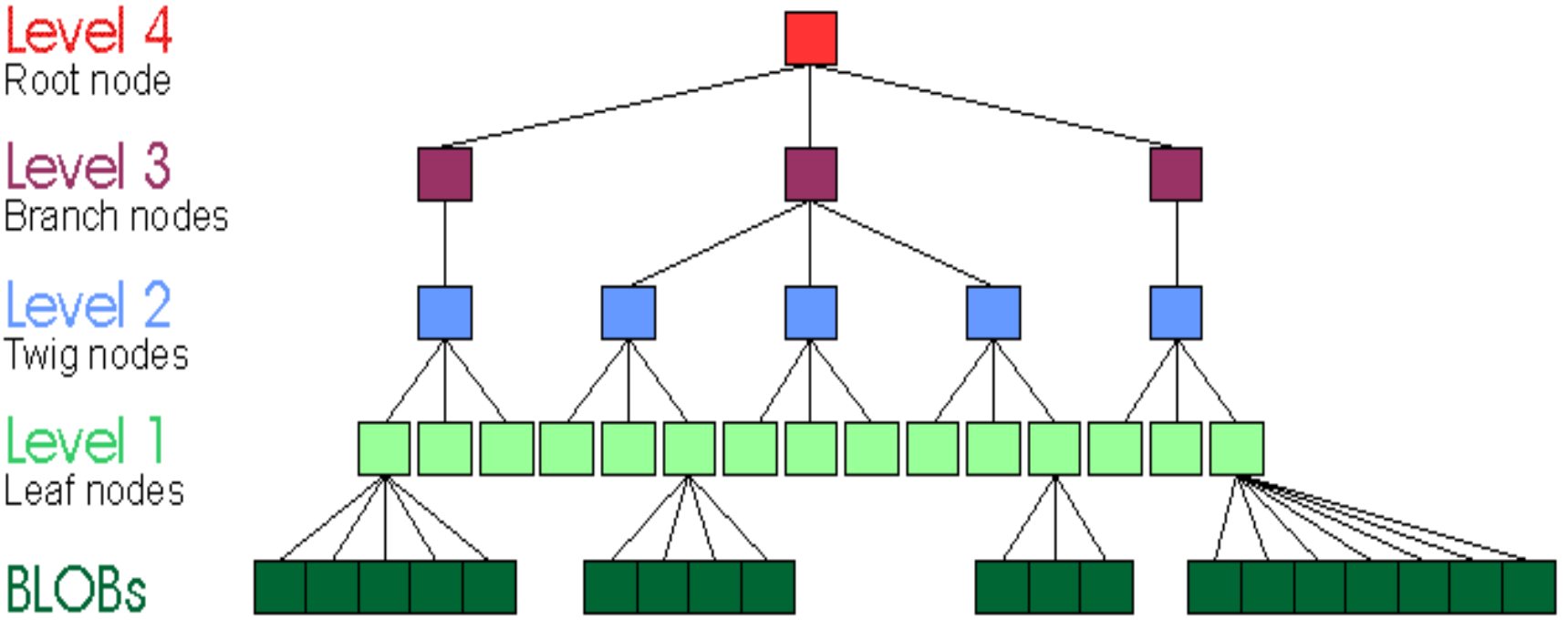
1. B+drzewo – zrównoważone drzewo o różnej liczbie gałęzi wychodzących z każdego wierzchołka, dane znajdują się tylko w liściach drzewa

2.Wymagana jest zajętość minimum 50% bloku, w innym przypadku dane są łączone w jeden węzeł

3. Szybkie wyszukiwanie plików w katalogu – czas logarytmiczny

B+drzewa c.d.

- Optymalizacje przy równoważeniu:
  - minimalizacja ilości użytych węzłów
  - minimalizacja ilości wyważanych węzłów
  - minimalizacja ilości wyważanych węzłów poza cache
  - maksymalizacja danych przenoszonych między sformatowanymi węzłami
- Wstawianie poprzez użycie **bitmapy wolnych bloków**, zaczynając od lewego sąsiada ostatnio używanego węzła i poruszając się w tym samym kierunku, co ostatnio – *stwierdzono eksperymentalnie, że metoda ta jest o 10% szybsza*



1. Węzły wewnętrzne (2, 3, 4) – wskaźniki i klucze (hashe nazw)
2. Liście – węzły sformatowane (metadane + pliki, ogony plików – reszta z podziału pliku na pełne bloki)
3. Węzły niesformatowane (BLOB) – tylko dane, całkowicie wypełnione, nie uwzględniane przy balansowaniu drzewa!

Węzeł sformatowany

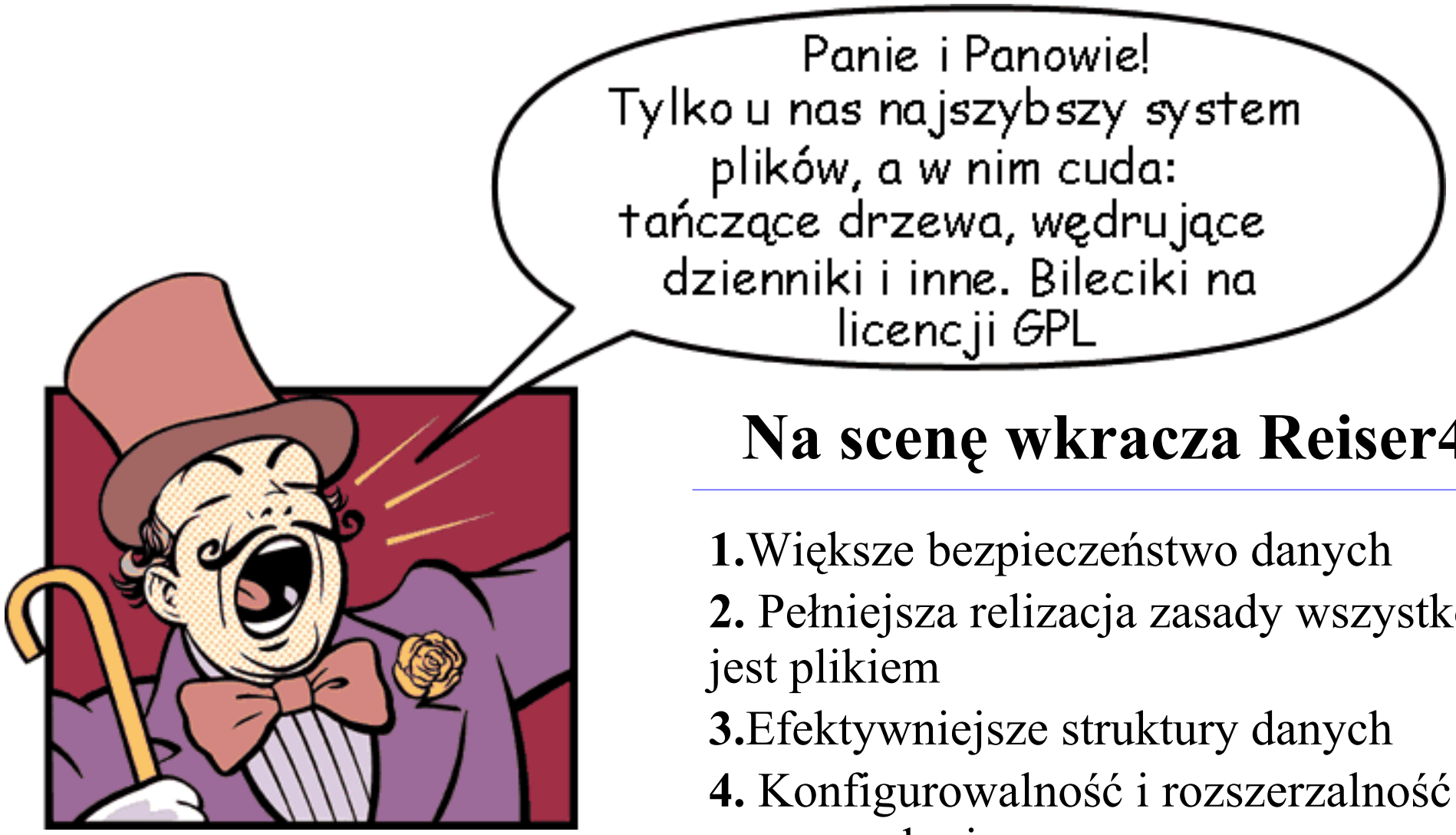
1. W węźle sformatowanym mogą być następujące typy danych - pozycje (item):

- dane katalogu
- metadane, atrybuty, typ pliku, rozmiar itp.
- dane bezpośrednie (całe małe pliki lub "ogony" dużych plików)
- dane pośrednie - wskaźniki na węzły niesformatowane

Upakowanie vs Wyrównanie

- Efektywne wykorzystanie miejsca
- Efektywniejsze operacje dyskowe
- Względna niezależność szybkości od wielkości bloku
- Potrzeba przepakowywania

- Marnotrastwo miejsca
- Mniejsza efektywność
- Zależne od wielkości bloku
- Prostota



Na scenę wkracza Reiser4

1.Większe bezpieczeństwo danych

2. Pełniejsza relizacja zasady wszystko jest plikiem

3.Efektywniejsze struktury danych

4. Konfigurowalność i rozszerzalność poprzez pluginy

5. Transakcyjność – ciąg operacji dyskowych jest kończony commitem i dopiero wtedy nanoszony na dysk

6. Lepsza obsługa dużych plików

Wędrujący dziennik

1.Normalnie kronikowanie wymusza dwukrotny zapis na dysk (dane i dziennik)

2. W Reiserze 4 zapisywany jest tylko dziennik, który po zapisie staje się częścią systemu plików, a dziennik wędruje w inne miejsce:

a) zapis w wolne miejsce

b) podmiana wskaźników na dane wskaźnikiem na dziennik

c) przejście w górę drzewa

Dziennik w Reiserze4

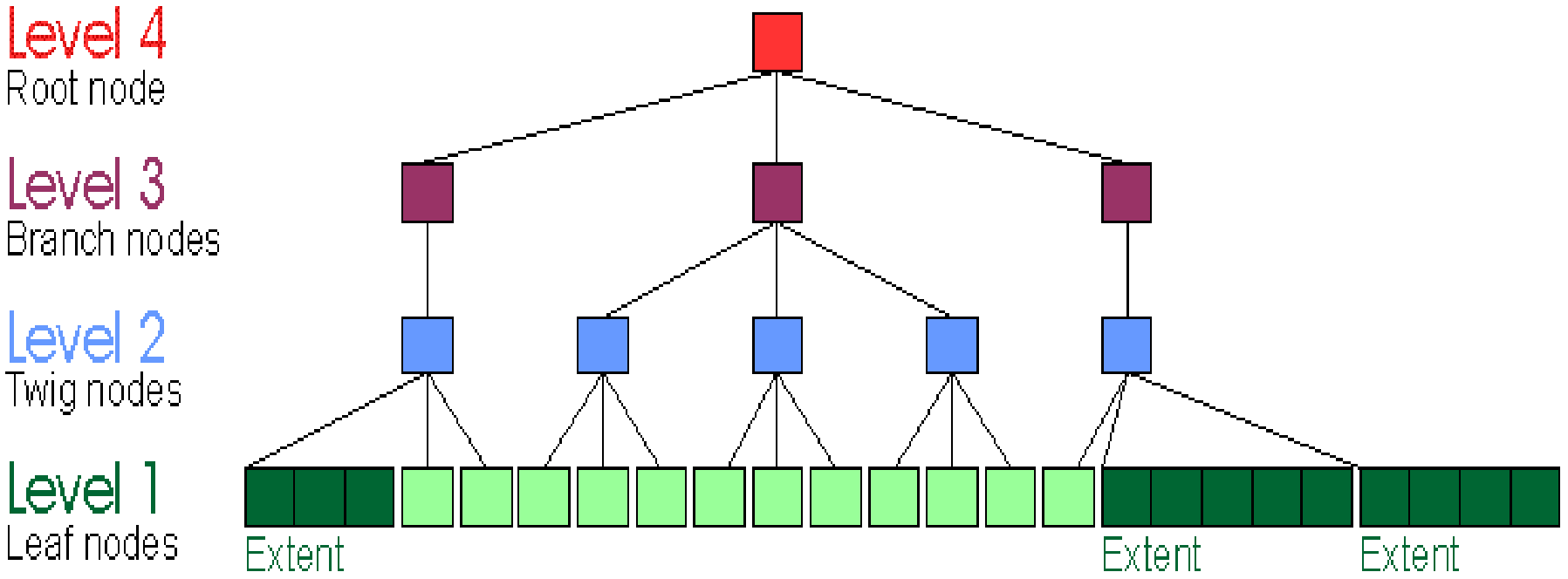
- Pomysł z przyspieszeniem kronikowania oparty na rozwiązaniu w systemie plików WAFL (Write Anywhere File Layout)
- Takie kronikowanie może prowadzić do fragmentacji plików – czasami może być optymalniejszym rozwiązaniem użycie standardowego kronikowania np.: zapisujemy środkowy fragment pliku, a potem odczytujemy cały plik wielokrotnie

Tańczące drzewa

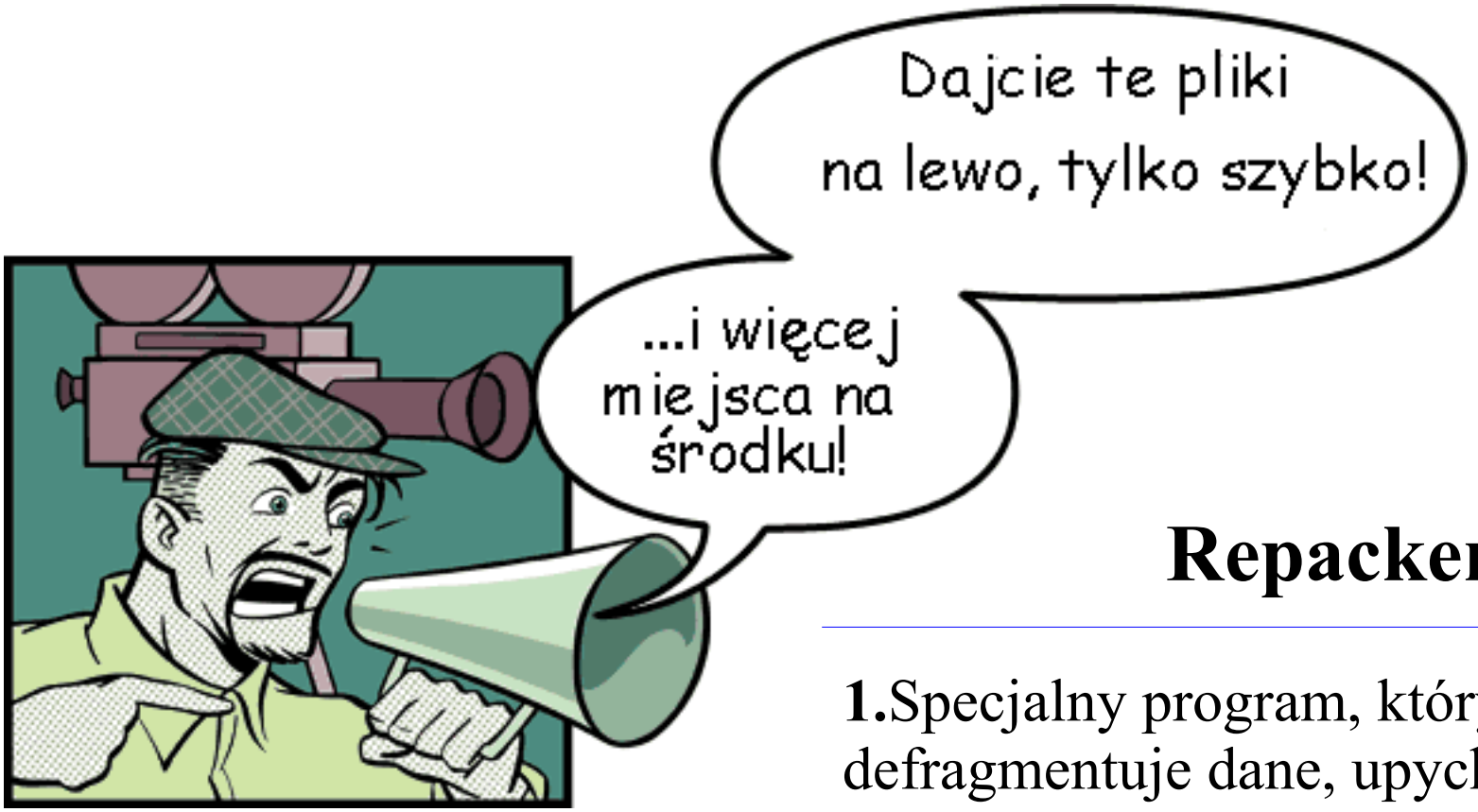
1.Nie dokonują scalania niepełnych wierzchołków przy każdej modyfikacji, a tylko przy commit'cie (zatwierdzeniu transakcji) – dlatego tańczące

2. Z eksperymentów wynika, że są szybsze

3. Extenty zamiast BLOBów – przedziały sąsiednich bloków należacych do jednego pliku (użyteczne dla dużych plików) – pomysl z XFS



- Extenty na poziomie liści (a więc są brane pod uwagę przy przebudowie drzewa!)
- Extent opisany jest poprzez wskaźnik i długość, a więc dla extentów dłuższych niż 2 bloki, ich opis jest mniejszy niż przy użyciu BLOBów z poprzedniej wersji

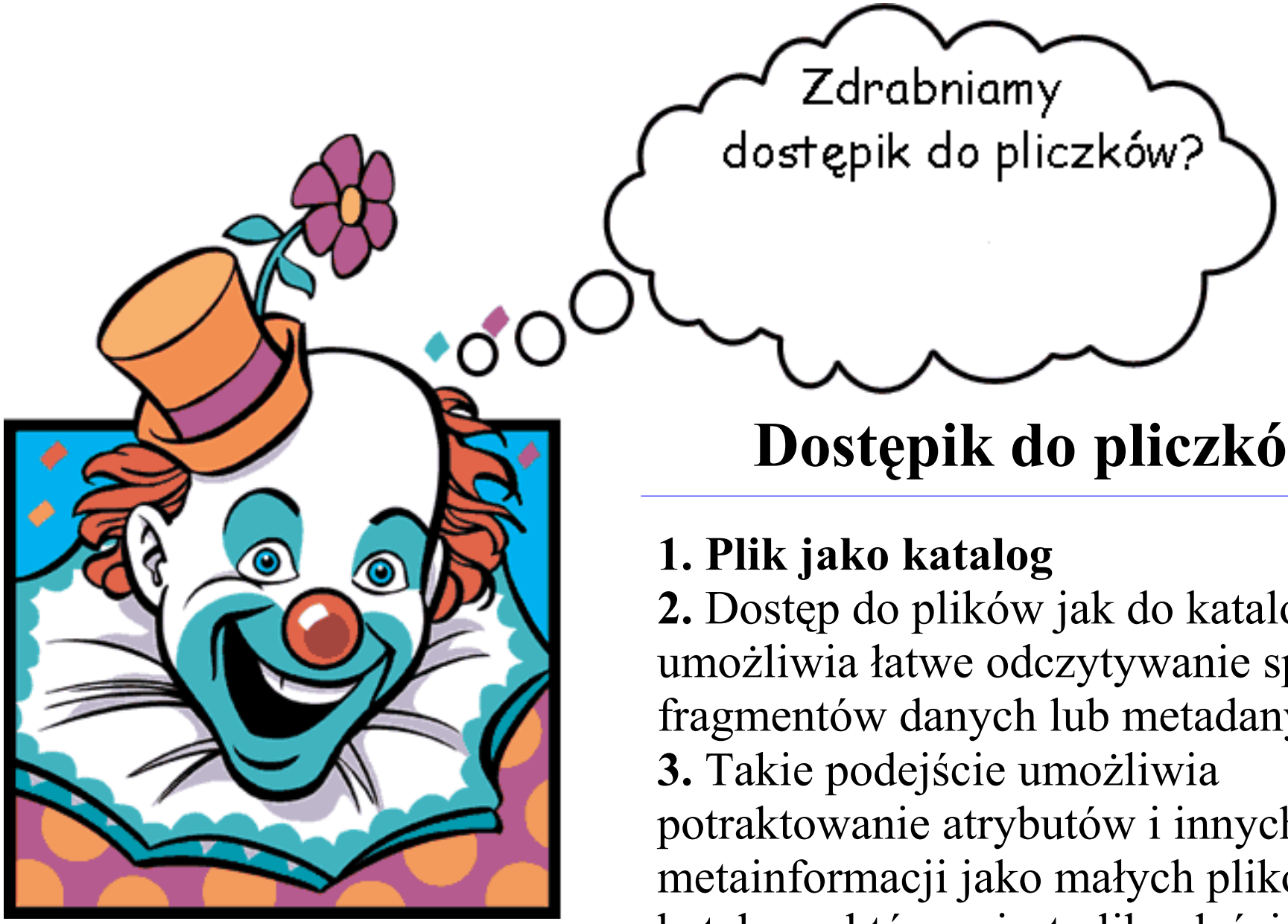


Repacker

1.Specjalny program, który porządkuje, defragmentuje dane, upycha ogony.
2. Sposób na radzenie sobie z fragmentacją, jaka jest wynikiem pakowania kilku plików do bloku i wędrującego dziennika
3. W przyszłości ma zamiast maksymalnego pakowania zostawiać “dziury”, żeby operacje wstawiania nowych węzłów były szybsze

Wszystko jest plikiem

1. Katalog jako plik
2. Katalog w Reiserze jest traktowany jako plik z listą plików
3. Dwie metody dostępu do plików:
a) zwykła
b) katalogowa (listująca zawartość)
3. Realizacja poprzez system wtyczek, który udostępnia plikom rożne metody dostępu do nich



Dostępik do pliczków

1. Plik jako katalog
2. Dostęp do plików jak do katalogów umożliwia łatwe odczytywanie spójnych fragmentów danych lub metadanych
3. Takie podejście umożliwia potraktowanie atrybutów i innych metainformacji jako małych plików w katalogu, którym jest plik właściwy!
4. Umożliwia to rozszerzanie listy atrybutów, łatwego udostępniania typu pliku MIME jako metadanych

Atrybuty jako pliki

- Przykład – zamiast ID3 Tag w plikach mp3, dostęp do tych informacji przez plik-katalog:
  - foo.mp3/artist
  - foo.mp3/title
- Takie podejście umożliwiłoby zwiększenie bezpieczeństwa np.: możnaby nadawać prawa dostępu tylko do części pliku – użyteczne w przypadku /etc/passwd:
  - /etc/passwd/501 – dane użytkownika 501

  w tej chwili atrybuty dotyczą całego pliku
- Duża ilość małych plików mogłaby być przeszkodą dla innych systemów plików, ale nie dla Reisera

Pluginy

1. System operacji na plikach jest rozszerzalny i modyfikowalny poprzez pluginy

2. Dzięki temu można łatwo dostosować Reisera do własnych potrzeb

3. Także niektóre problemy – jak kolizje przy haszowaniu nazw plików mogą być rozwiązane w przyszłości przez zastosowanie pluginów

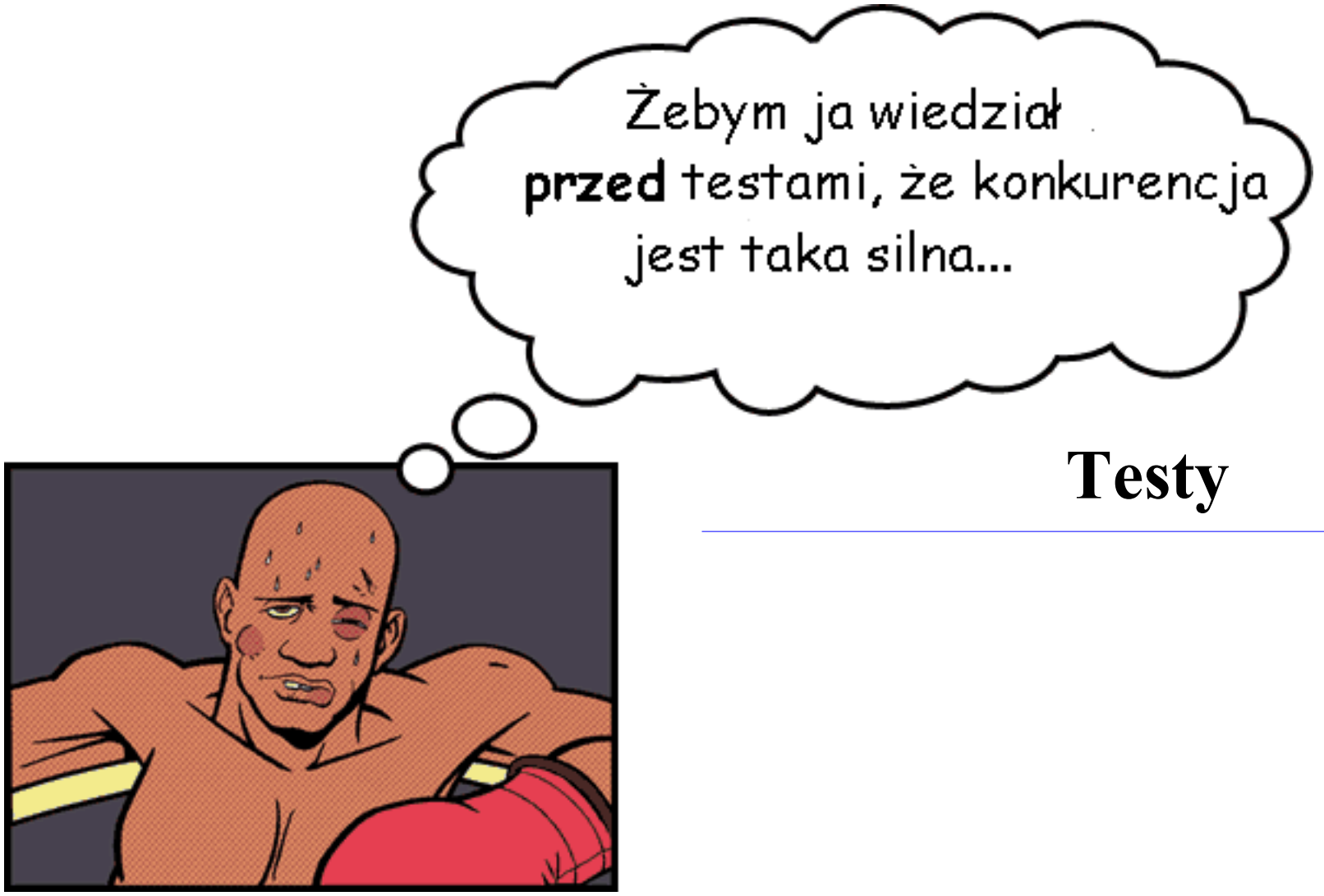
4. W tej chwili po zmianie pluginów potrzebna jest rekompilacja jądra

Rodzaje wtyczek

* **file plugins** - dostęp do plików
* **directory plugins** - dostęp do tradycyjnych katalogów
* **hash plugins** - haszowanie kluczy w drzewie tańczącym
* **security plugins** - bezpieczeństwo
* **item plugins** - dostęp do pozycji (item) w węzłach sformatowanych
* **key assignment plugins** - zajmują się przydzielaniem kluczy w drzewie
* **node search and item search plugins** - odpowiedzialne za wyszukiwanie w drzewie węzłów i pozycji (item)

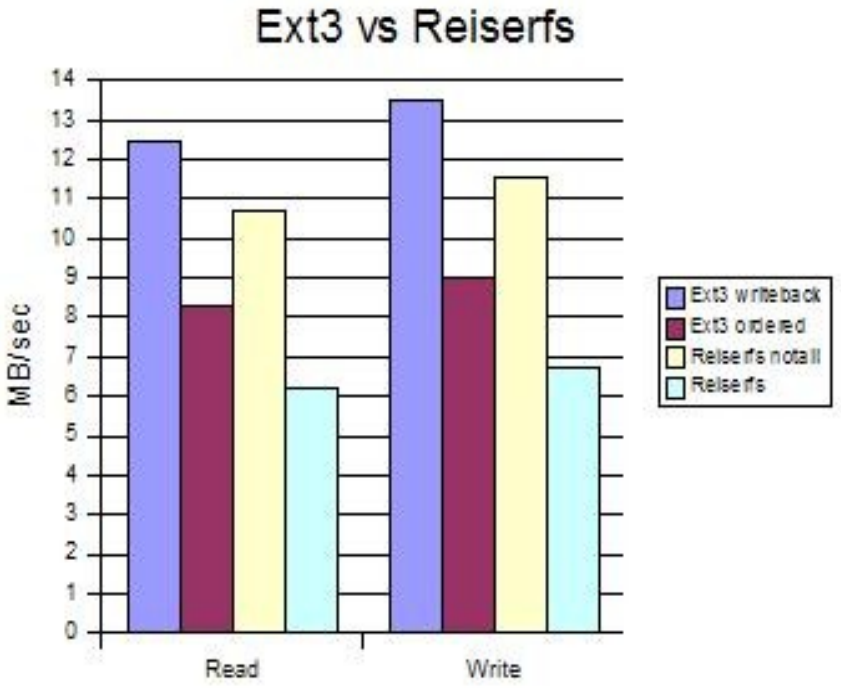
Reiser 5 i 6

Reiser 5 – rozproszony system plików
Reiser 6 – system o rozszerzonej semantyce, zamiast drzewiastej struktury plików, dowolny graf
-wolniejszy, ale bardziej intuicyjny
-możliwość porządkowania plików według różnych kryteriów

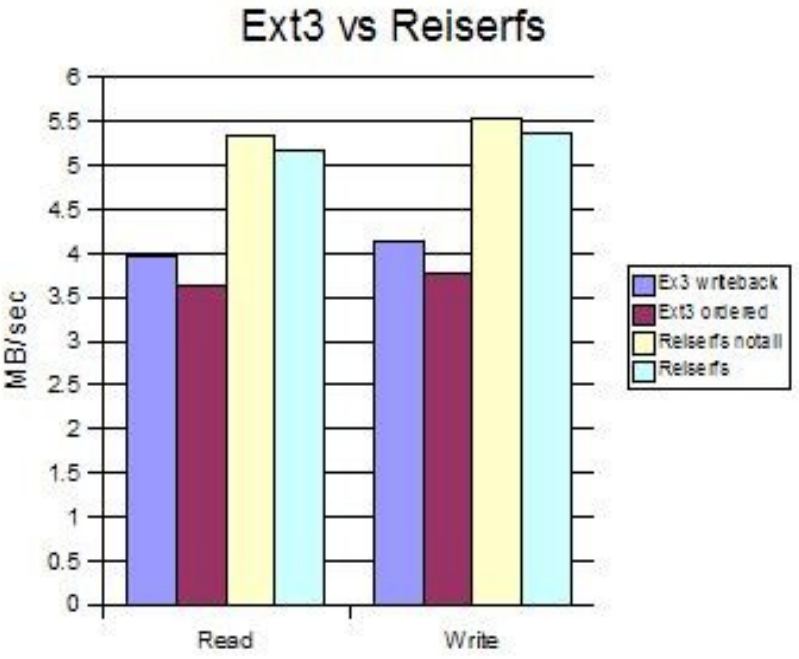

Żebym ja wiedział
przed testami, że konkurencja
jest taka silna...

Testy

Ext3 i ReiserFS



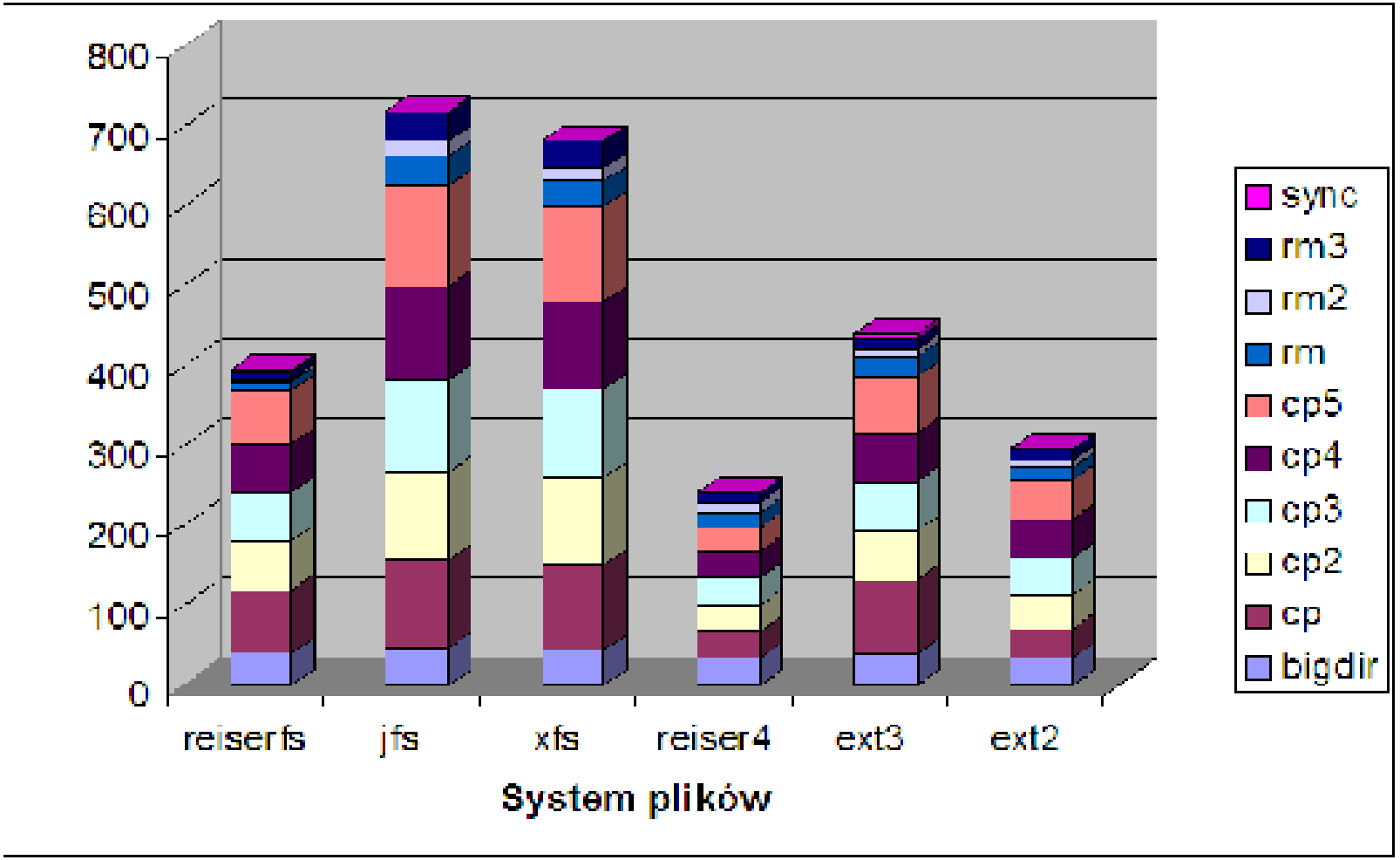
Pliki od 1KB do 9KB
mały ruch (150MB w sumie)



Pliki do 300KB
duży ruch (ponad 19GB)

Przeprowadzone przez firmę Guru Labs

Reiser na tle różnych systemów



Wewnętrzna fragmentacja

System	Fragmentacja wewnętrzna
Reiser	6%
Reiser notail	14%
XFS	15%
JFS	17%
ext3	21%

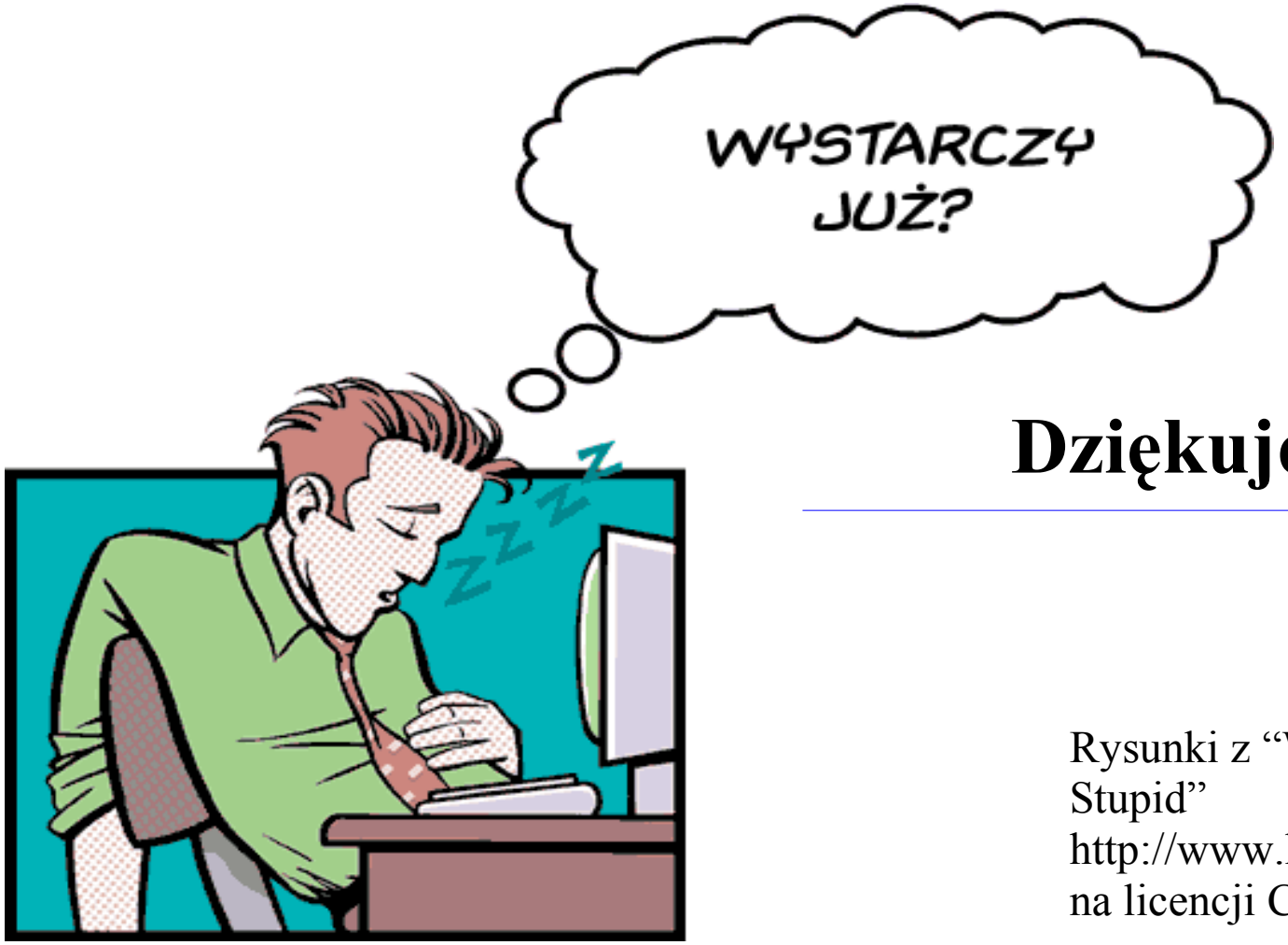
Wykonany przez Constantina Loizidesa z Uniwersytetu we Frankfurcie na partycji 4GB

Podsumowanie cech Reisera4

- Nowoczesny system plików:
  - Obsługa dużych partycji (do 16EB) i plików (do 8EB)
  - Bezpieczeństwo danych: kronikowanie, system tworzony min. do zastosowań wojskowych
- Zdobywający coraz większa popularność
- Możliwość rozszerzenia poprzez wtyczki o funkcjonalności konkurentow (szyfrowanie, kompresja), ktorych standardowo brak

Podsumowanie

- Łączący najlepsze rozwiązania z innych systemów i własną innowacyjność:
  - Szybkie, wędrujące dzienniki
  - B+drzewa z dobrą obsługą małych plików
  - Extenty z polepszoną obsługą plików dużych
  - Pluginy
  - Rozszerzalność atrybutów
  - Konsekwencja w stosowaniu zasady wszystko jest plikiem



Dziękuję za uwagę

Rysunki z “Why Tables for Layout is Stupid”
http://www.hotdesign.com/seybold/
na licencji Creative Commons

Użyteczne informacje:

- Strona Reisera 4: http://www.namesys.com/v4/v4.html
- Porównanie systemów plików
http://en.wikipedia.org/wiki/Comparison_of_file_systems
- Prezentacje z lat poprzednich na stronie SO