

### Zadanie 1.

Według artykułu problemem w Unixach jest to, że plik jest zwykłym ciągiem bajtów, bez żadnej innej informacji. Ma to taki skutek, że do obsługi hardware (bo też jest traktowane jako device - plik, czyli ciąg bajtów) trzeba używać brzydkiego interfejsu `ioctl`, czy robić jakieś hacki do jądra. Kolejnym problemem jest to, że programy piszące do plików muszą coś wiedzieć o ich strukturze. Jedynym sposobem na zawarcie metadanych jest umieszczenie ich w tym ciągu bajtów.

W MacOS obeszli to w ten sposób, że każdy plik może zawierać część **data fork**, czyli właściwe, zwykłe dane oraz **resource fork**- sekcje do ustrukturyzowanych danych. Przykładowo **resource fork** może przechowywać informację o ikonie pliku, kształty okienek czy zawartość jakichś menu. Inny przykład- dokument tekstowy. Treść tekstowa dokumentu może być trzymana w **data fork**, a wszystkie obrazki tam umieszczone w **resource fork**. Istnieje masa predefiniowanych typów sekcji w **resource fork**: ogólne info o aplikacji, ikony, paleta kolorów, machine code, kształt kursora, itp.

Okazuje się, że od pewnej, starej wersji Linuxa istnieją **rozszerzone atrybuty pliku**. Są to pary w postaci klucz:wartość, które są powiązane z pewnym plikiem lub katalogiem. Ich przechowywanie jest ściśle uzależnione od systemu plików. W `ext3` jest dla nich specjalna sekcja w `inode` i mogą być trzymane w osobnym bloku.

Flaga `xattr` w `wget` dołącza dodatkowe informacje o ściągniętym pliku jako rozszerzone atrybuty, np. adres URL.

```
wget --xattr http://example.com/
```

Można teraz sprawdzić atrybuty tego pliku:

```
getfattr -d index.html
```

Wynik:

```
# file: index.html
user.xdg.origin.url="http://example.com/"
```

Warto wspomnieć, że w nazwy atrybutów mają postać `namespace.attribute`, gdzie `namespace` jest jednym z `security`, `system`, `trusted`, `user`.

Sumę md5 możemy przypisać do atrybutu poleceniem:

```
setfattr -n user.md5sum -v $(md5sum index.html | cut -c -32) index.html
```