

Wstęp do Informatyki dla Bioinformatyków

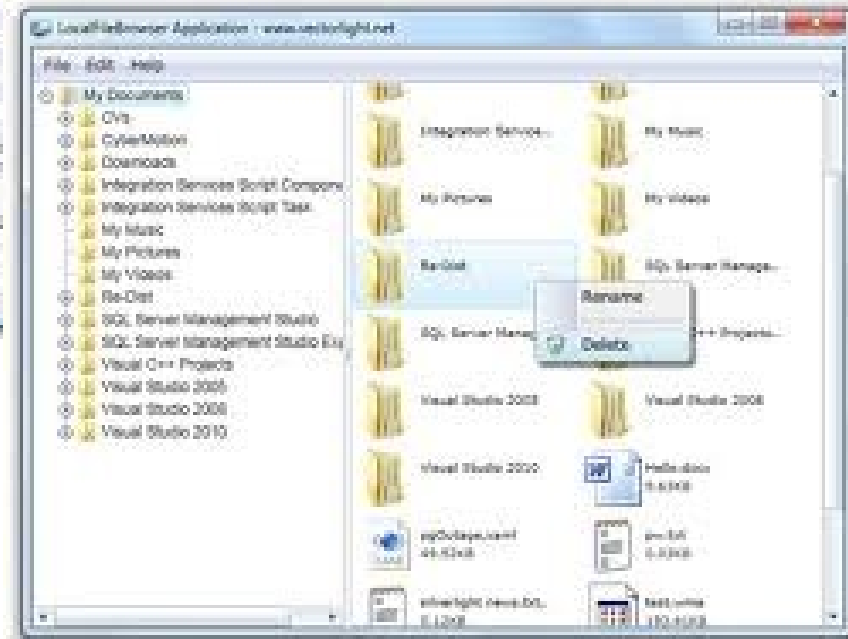
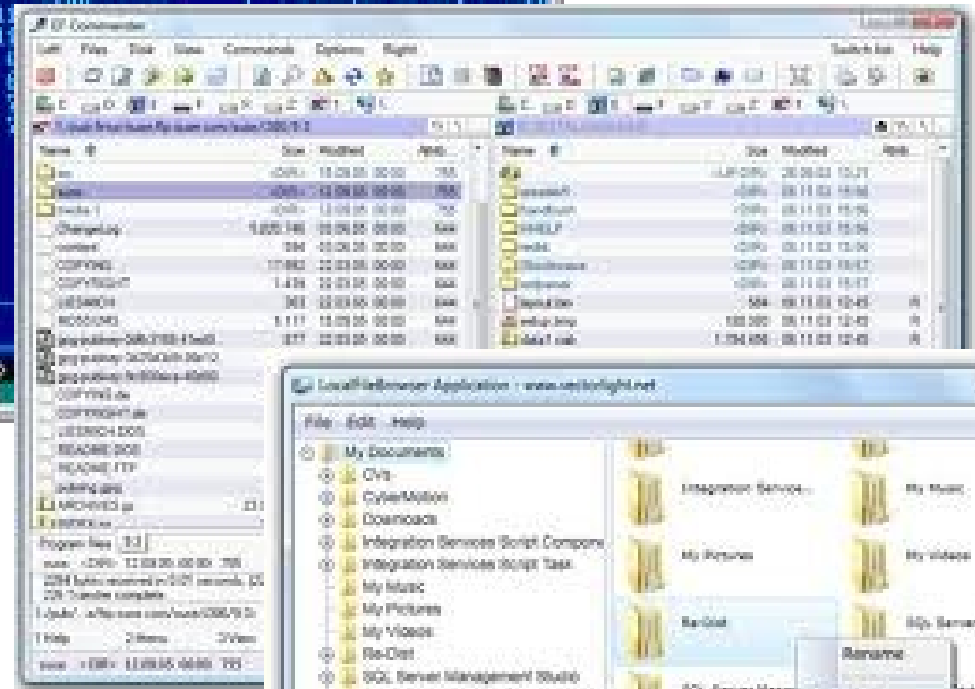
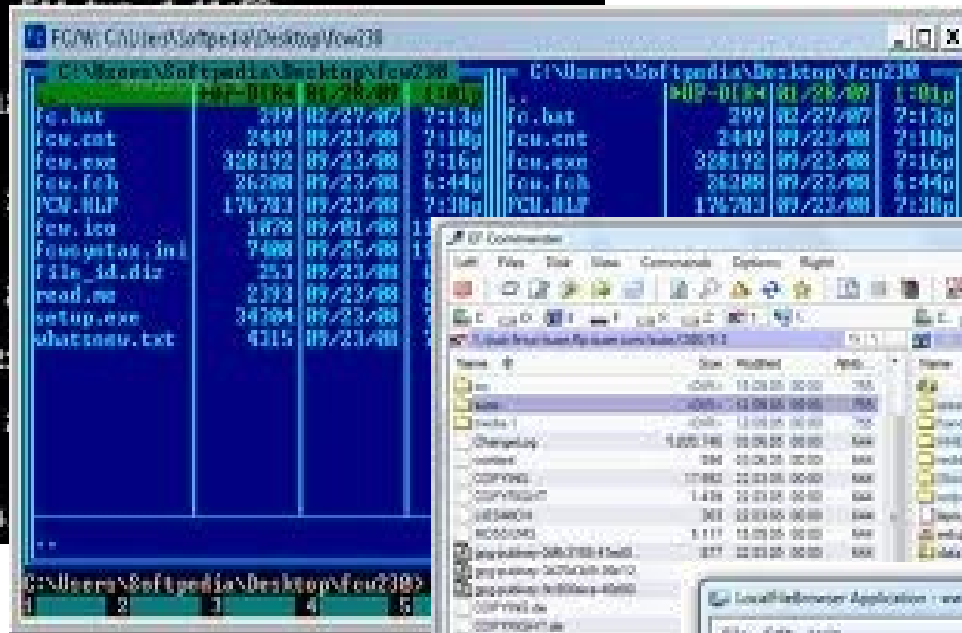
Wykład 4: Pliki

Bartek Wilczyński

26 X 2015

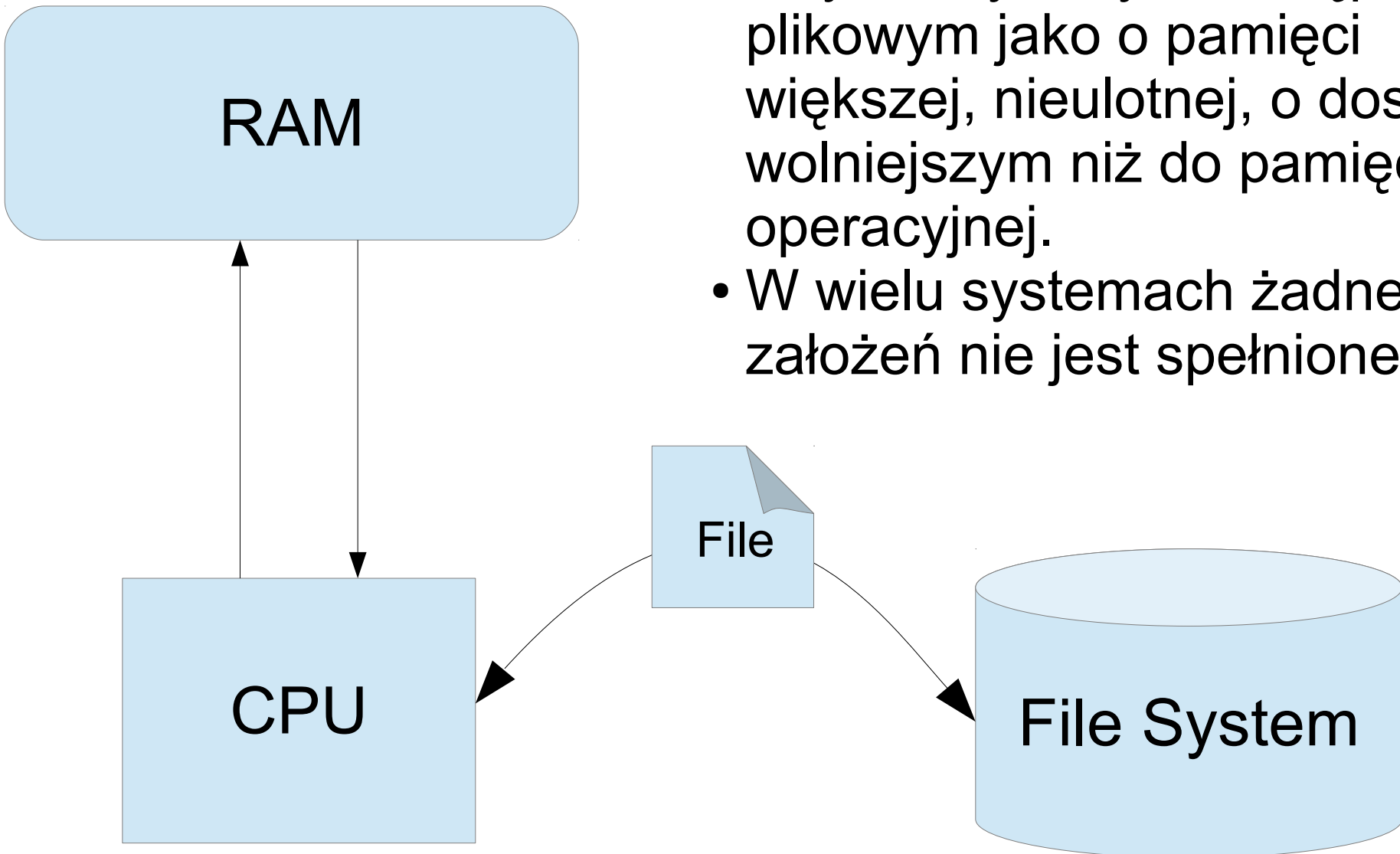
Pliki okiem użytkownika

```
Flavio's-MacBook-Pro:redis-2.4.16 Flavio$ ll
total 72
drwxr-xr-x  16 Flavio staff
drwx-----+ 14 Flavio staff
-rw-r--r--   1 Flavio staff 1
-rw-r--r--   1 Flavio staff 1
-rw-r--r--   1 Flavio staff 1
-rw-r--r--   1 Flavio staff 1
-rw-r--r--   1 Flavio staff 1
-rw-r--r--   1 Flavio staff 1
-rw-r--r--   1 Flavio staff 1
-rw-r--r--   1 Flavio staff 1
-rw-r--r--   1 Flavio staff 1
drwxr-xr-x   5 Flavio staff
-rw-r--r--   1 Flavio staff 2
-rw-r--r--   1 Flavio staff 1
drwxr-xr-x  74 Flavio staff
drwxr-xr-x   9 Flavio staff
drwxr-xr-x   8 Flavio staff
Flavio's-MacBook-Pro:redis-2.4
```



Pliki w maszynie von Neumanna

- Zwykle myślimy o dostępie plikowym jako o pamięci większej, nieulotnej, o dostępie wolniejszym niż do pamięci operacyjnej.
- W wielu systemach żadne z tych założeń nie jest spełnione



Plik okiem programisty

```
Ala ma kota.▼  
Kot ma ale▼  
>Human Chromosome 1▼  
NNNNNNTAACCCCTAACCCCTAACCC  
TAACCCTAACCCCTAACCCCTAACCC  
TAACCCTAACCCCTAACCCCTAACCC  
TAACCCTAACCCCTAACCCCTAACCC  
TAACCCTAACCCCTAACCCAACCCT  
AACCCCTAACCCCTAACCCCTAACCCCT  
AACCCCTAACCCCTAACCCCTAACCC  
TAACCCTAACCCCTAACCCCTAACCCCT  
AACCCCTAACCCCTAACCCCTAACCC  
TAACCCTAACCCCTAACCCCTAACCC  
CCTAACCCCTAACCC■AACCCCTAAC  
CCTAACCCCAA●
```

- Spójny ciąg bajtów (tekstowy lub binarny)
- Znaki końca linii (EOL), końca pliku (EOF)
- Kursor, który można przesuwać
- Możliwość zapisu i odczytu
- Nie wszystkie operacje dla plików binarnych

Pliki w Pythonie

- Otwieranie plików

```
>>> f = open('/tmp/workfile', 'w')
>>> print f
<open file '/tmp/workfile', mode 'w' at 80a0960>
```

- Możliwe tryby odczytu:
 - r,rb,r+ :do odczytu
 - w,wb,w+ :do zapisu
 - a,ab,a+ : do “dopisywania”

Czytanie z pliku

- Różne sposoby czytania z pliku:

```
>>> f.read()
'This is the entire file.\n'
>>> f.read()
''
```

```
>>> f.readline()
'This is the first line of the file.\n'
>>> f.readline()
'Second line of the file\n'
>>> f.readline()
''
```

```
>>> f.readlines()
['This is the first line of the file.\n', 'Second line of the file\n']
```

Zapisywanie danych do pliku:

- Zapisywanie napisów

```
>>> f.write('This is a test\n')
```

- Przekształcanie innych wartości na napisy

```
>>> value = ('the answer', 42)
>>> s = str(value)
>>> f.write(s)
```

- Znaki specjalne: “\n”, “\r”, “\t”...

Przeglądanie plików

- Iterowanie “linia po linii”

```
>>> for line in f:  
    print line,
```

```
This is the first line of the file.  
Second line of the file
```

- Wyszukiwanie

```
>>> f = open('/tmp/workfile', 'r+')  
>>> f.write('0123456789abcdef')  
>>> f.seek(5)      # Go to the 6th byte in the file  
>>> f.read(1)  
'5'  
>>> f.seek(-3, 2) # Go to the 3rd byte before the end  
>>> f.read(1)  
'd'
```

- Przydatna funkcja `f.tell()`

Zamykanie plików

- Aby uniknąć utraty zapisanych danych, należy zamykać pliki.
- Można ręcznie:

```
>>> f.close()
>>> f.read()
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
ValueError: I/O operation on closed file
```

- Albo automatycznie:

```
>>> with open('/tmp/workfile', 'r') as f:
...     read_data = f.read()
>>> f.closed
True
```

Słowo o serializacji

- Jeśli nie chcemy zajmować się szczegółami reprezentacji obiektów python'a, możemy nasze obiekty “peklować” (`import pickle`)
- Założmy, że mamy otwarty plik `f`
- Możemy tam “zapeklować” dowolny obiekt `x`:

```
pickle.dump(x, f)
```

- I później go “odpeklować” (choćby na innym komputerze):

```
x = pickle.load(f)
```

Dodatek o wyszukiwaniu binarnym

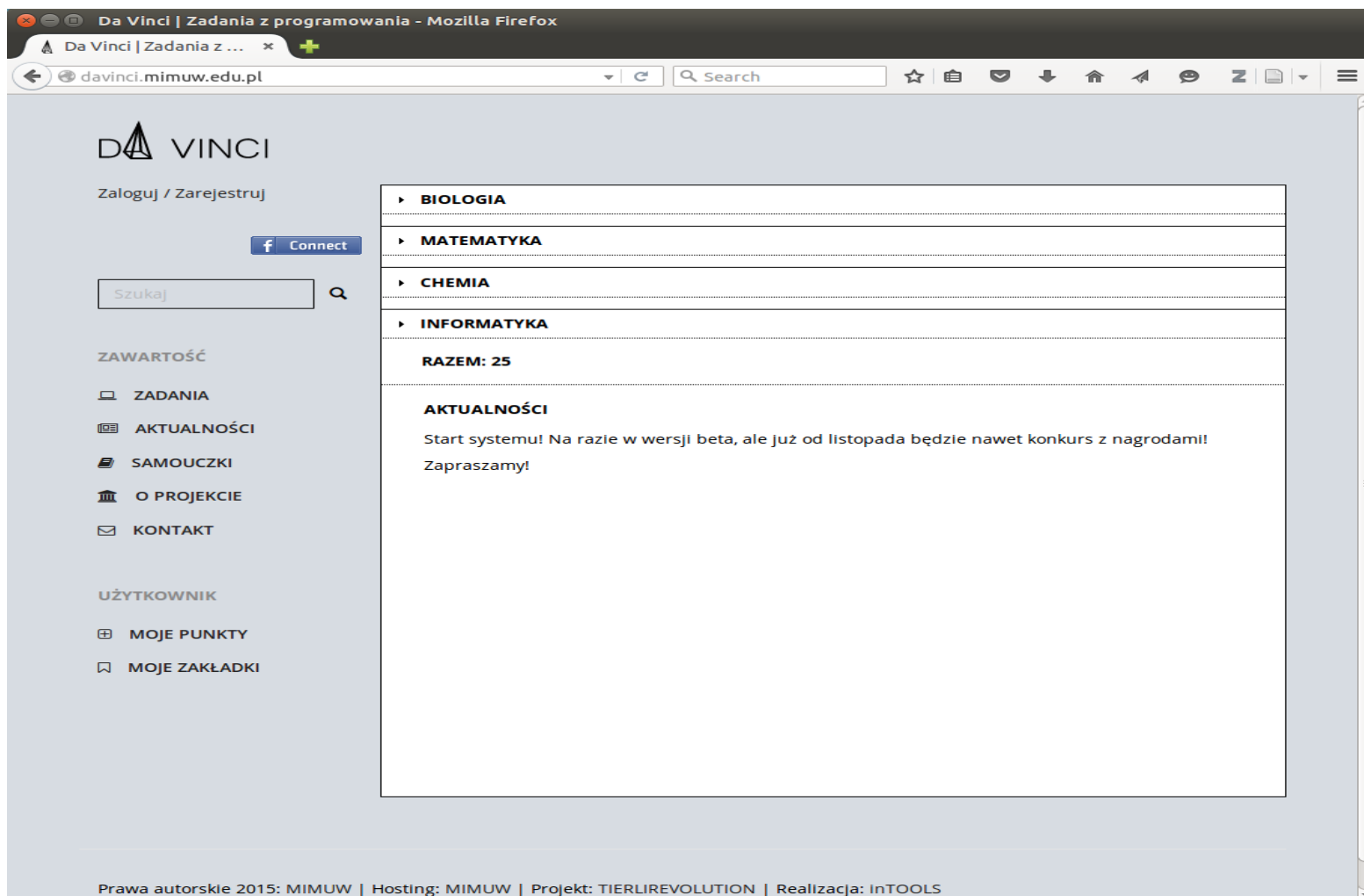
- Jak rekurencyjnie znaleźć wartość w posortowanej tablicy?
- Wersja naiwna: przejrzeć wszystkie elementy...
- Wersja sprytniejsza: Sprawdźmy wartość w połowie, i rozwiążmy zadanie rekurencyjnie...

Tematy na Kolokwium (3.11)

- Świadome używanie typów danych (liczby, napisy, listy, porównywanie, działania)
- Proste zadania iteracyjne (wyszukiwanie, podsumowywanie, wyliczanie iteracyjne wartości)
- Przystawianie elementów w kolekcjach (np. Sortowanie bąbelkowe)
- Definiowanie funkcji, również rekurencyjnych (np. rekurencyjny anagram)

Projekt davinci - zadania z programowania

<http://davinci.mimuw.edu.pl>



The screenshot shows the website 'Da Vinci | Zadania z programowania' in a Mozilla Firefox browser. The address bar displays 'davinci.mimuw.edu.pl'. The page layout includes a header with the 'DA VINCI' logo and a navigation menu on the left. The main content area features a list of subjects: BIOLOGIA, MATEMATYKA, CHEMIA, and INFORMATYKA, with a total count of 25. Below this is a news section titled 'AKTUALNOŚCI' with a message about the system's beta version and an upcoming competition.

Da Vinci | Zadania z programowania - Mozilla Firefox

Da Vinci | Zadania z ...

davinci.mimuw.edu.pl

DA VINCI

Zaloguj / Zarejestruj

Connect

Szukaj

ZAWARTOŚĆ

- ZADANIA
- AKTUALNOŚCI
- SAMOU CZKI
- O PROJEKCIE
- KONTAKT

UŻYTKOWNIK

- MOJE PUNKTY
- MOJE ZAKŁADKI

► **BIOLOGIA**

► **MATEMATYKA**

► **CHEMIA**

► **INFORMATYKA**

RAZEM: 25

AKTUALNOŚCI

Start systemu! Na razie w wersji beta, ale już od listopada będzie nawet konkurs z nagrodami!
Zapraszamy!

Prawa autorskie 2015: MIMUW | Hosting: MIMUW | Projekt: TIERLIREVOLUTION | Realizacja: InTOOLS