

Лекция 10 (06. Декември, 2011)

Първи час

Допълнения към C-style четене и писане (има ги в лекция 6)

```
mode ('r', 'w', 'a'). "r+", "w+", "rt", "rb".  
fclose()  
int fscanf(FILE* stream, const char* format, ... );  
fgets(char* destination, int maxNumberOfReadCharacters, FILE* source);  
feof(FILE* stream)
```

```
int fscanf(FILE* file, const char* format, ...);  
int fprintf(FILE* file, const char* format, ...);
```

Пренасочване на входа и изхода (има ги в лекция 6)

```
freopen(const char* fileName, const char* mode, FILE* stream);  
freopen("inputFile.txt", "rt", stdin);  
freopen("outputFile.txt", "wt", stdout);
```

Алтернативи на bug-tracking системите (има ги в лекция 8)

Какви алтернативи можем да ползваме вместо Bug-tracking системи? Какви са техните предимства и недостатъци? "Външни" и „вътрешни“ bug-tracking системи.

Разлика между компилирани и интерпретирани езици. Защо Java е по-бавна.

- Много накратко какво е машинен код и какво е асемблер – машинен код е поредица от процесорни инструкции (представени чрез техните уникални идентификационни кодове) и потенциално аргументи, които процесорът може да изпълни. Асемблер е почти същото нещо, само че вместо да се ползват само числа, някои от тях (предимно инструкциите) са заместени с кодови думи (тоест вместо уникални числа се ползват уникални думи за различните инструкции). Това е значително по-удобно за разбиране от човек. Съответно езикът Асемблер може да бъде (и дълго време е бил и дори сега е) ползван от програмисти. Това е език от много ниско ниво – неговите инструкции директно се изпълняват от хардуера, докато при повечето езици, които познаваме, както ще видим това не е така. Пример за машинен код е ехе-то, което се създава, когато компилирате своята C/C++ програма.
- Компилираните езици, са програмни езици, чиито код не се изпълнява директно от компютъра. Преди изпълнението на дадена програма, нейният source code трябва да се преведе до машинен код (или асемблер), като след това се изпълняват директно процесорни инструкции. Компилацията понякога може да е бавна (особено при големи проекти), затова има различни начини за нейното забързване. Един от тези начини, които много накратко разгледахме на лекцията е програмата make.
- От друга страна интерпретирани езици не се нуждаят от предварителна трансляция – техният код се изпълнява стъпка по стъпка от така наречените интерпретатори. Има няколко предимства за ползване на некомпилirани езици – например няма нужда от компилиране; кодът им е Platform-

independent (стига да има интерпретатор за въпросната платформа); разрешават ползване на динамично-типизирани променливи.

- Самото парсване на source code-а може да е бавно (явен пример за това са PHP и Javascript), затова в някои случаи се извършва комбинация между компилиране и интерпретиране. Началният сорс се транслира до междинен език (който не е нито първоначалния сорс код нито машинен език – така наречения bytecode), който в последствие бива интерпретиран. Най-известният пример за това, разбира се, е Java, където входният сорс се компилира до class file-ове (това е bytecode-ът), които в последствие се интерпретират по време на изпълнението от Java Virtual Machine.
- Примери за компилирани езици: C, C++, Pascal, Scheme, Haskell, Ada, Delphi, Fortran, ALGOL.
- Примери за интерпретирани езици: PHP, Ruby, Perl, ASP, Mathematica, MATLAB, повечето скриптов езици.
- Примери за смесени (компилятор + интерпретатор) езици: Java, Python, C#
- Статистики

<http://lang-index.sourceforge.net/>

Any			General Purpose		
Rank	Name	Share	Rank	Name	Share
1	Java	18.652%	1	Java	26.336%
2	C	17.719%	2	C	25.019%
3	Objective-C	9.719%	3	Objective-C	13.723%
4	C++	6.400%	4	C++	9.036%
5	PHP	6.179%	5	Basic	7.860%
6	Basic	5.567%	6	C#	4.900%
7	Python	3.804%	7	Delphi	1.853%
8	Perl	3.766%	8	D	1.708%
9	C#	3.470%	9	Ada	1.424%
10	JavaScript	2.225%	10	Pascal	1.279%

Ползване по състезания

<http://www.go-hero.net/jam/11/languages/0>

C++ над два пъти по-използван от Java

<http://community.topcoder.com/tc?module=ProblemDetail&rd=14550&pm=11665>

C++ над пет пъти по-използван от Java

Миграция от Java към C в мобилните платформи

Втори час

Задачи от интервюта