

PODZIMNÍ SOUSTŘEDĚNÍ KSP 2015 – SEZNAM PŘEDNÁŠEK

Tento spisek jest nabídkou přednášek, které byste na soustředění mohli slyšet, čili jakási obdoba matfyzácké Karolínky (ta je ale, pravda, ještě stále o něco tlustší). Přednášek je daleko víc, než kolik se dá za pár dní stihnout, a tak je na vás, abyste si vybrali, o které máte opravdu zájem; pokud byste rádi slyšeli ještě o něčem dalším, klidně si o to napište (třeba na fórum), třeba se najde někdo, kdo by vám o tom rád pověděl. Berte a vychutnávejte!

Údaje o jedné přednášce vypadají asi takto:

Stručný úvod do základů teorie vlkodlaků (“*Za dne ukryt v hloubi lesa, děs temný zvečera se plazí. . .*”) **LYK**

RNDr. Á. Cula

Úvod do moderní teorie vlkodlaků, čili též praktická dæmonologie a naiadologie.

Předpoklady: Měsíc v úplňku.

Dozvíte se (čteno v obvyklém pořadí): jméno přednášky, v uvozovkách motto přednášky, kód (pro snadnější odkazování na konkrétní předměty), jméno přednášejícího a nakonec stručný obsah přednášky. Hvězdičky znamenají obtížnost.

Základní přednášky

V této kategorii sídlí přednášky, které se dají považovat za základní stavební kameny informatiky, ať teoretické, či praktické.

Algoritmy a datové struktury

Algoritmy a jejich složitost (“*Čím menší je časová složitost algoritmu, tím větší je složitost kódu.*”) **SLOZ**

Problém, algoritmus a program. Časová a paměťová složitost problémů i algoritmů. Složitost rekurzivních algoritmů, složitost v průměrném případě. Ukázky jednoduchých (obvykle aritmetických a třídicích) algoritmů a výpočet jejich složitosti.

Základní algoritmy

ZALG

Základní výbavou každého informatika jsou různé standardní algoritmy, zde si ukážeme ty nejdůležitější z nich. Třídicí algoritmy – porovnávací i přihrádkové. Hledání k -tého nejmenšího prvku v lineárním čase. Práce s výrazy a železničářský algoritmus.

Grafy & algoritmy (“*Pojďme si hrát s obrázky*”) **GA**

Co to jsou grafy, jak je v programech reprezentovat a hlavně k čemu se dají použít. Prohledávání grafu do šířky i do hloubky. Hledání nejkratších cest: Dijkstrův a Floydův algoritmus. Minimální kostry a Union-Find problem.

Prohledávání do hloubky

DFS

Karry Burešová

Trochu hlubší pohled na prohledávání do hloubky. Jeho (často dost nečekané) aplikace v dalších algoritmech, jako je třeba hledání mostů, topologické třídění, rozklad na komponenty silné souvislosti či kreslení grafu jedním tahem.

Toky v sítích (“*Když je v grafu povodeň, těsní?*”) **TOKY**

Jirka Setnička

K čemu je dobré, když grafem teče voda. Předvedeme si klasický problém toků v sítích a jeho všelijaké, mnohdy dosti překvapivé aplikace. Jak rozestavět n věží na šachovnici a jak ji místo toho pokrýt dominovými kostkami? Další souvislosti, jako třeba násobná souvislost grafů.

Předpoklady: Umět plavat (zejména v matematice)

Datové struktury pro začátečníky (“*Pole oraná a neoraná, stromy ovocné a okrasné.*”) **DS1**

Jak si ukládat data natolik šikovně, abychom je nejen neztratili, ale také našli dříve, než si pro nás přijde Smrť. Klasické struktury jako pole, seznamy, fronta a zásobník, trie, vyhledávací stromy (vyvážené, AVL, $a-b$, splay), haldy (binární a obecně regulární) a v neposlední řadě hešování.

Datové struktury pro pokročilé * (“*Haldy a jiné kupky.*”) **DS2**

Jirka Setnička

Důmyslnější varianty vyhledávacích stromů: splay stromy, $BB-\alpha$ stromy, vícerozměrné stromy. Chytřejší haldy: binomiální, Fibonacciho, 2-3. Amortizovaná analýza složitosti. Též několik přátelských randomizovaných datových struktur: skip listy a treapy.

Intervalové stromy * (“*Já bych ty intervaly nejradši. . . dal do stromu!*”) **ITREE**

Jirka Setnička, Janka Bátoriová

Intervalový strom je datová struktura pracující s intervaly, se kterou se můžeme setkat v mnoha úlohách (zejména soutěžních). Řekneme si, co to intervalový strom je, jaké všechny druhy intervalových stromů existují a jejich použití si ukážeme na úlohách. Na závěr si představíme jednu „magickou“ datovou strukturu jménem Fenwickův strom.

Dynamické programování (*“Kampak jsem si to jenom schoval?”*) **DYNP**

Janka Bátoryová, Martin Šerý

Dynamické programování je programátorská technika využívající velice prostinkého nápadu: Proč něco počítat několikrát, když to mohu spočítat jednou a výsledek si uložit? Na této přednášce si ukážeme, že tento jednoduchý nápad může pomoci efektivně vyřešit i poměrně obtížné úlohy.

Hledání v textu (*“»Vyšíváme v seníku!« – kde jsem to jen viděl?”*) **REGEX**

Jirka Setnička, Filip Štědranský

Někdy potřebujeme najít podřetězec ve velkém množství textu. Stromeček trochu připomínající ten biologický aneb trie. Proč se ve vstupu vracet neboli Knuthův-Morrisův-Prattův algoritmus. Hledání více řetězců najednou podle Aha a Corasickové. Okénkové hešování Rabina a Karpa.

Parsing čili analýza textu (*“ $1+2*4 = 12$ ”*) **PARSE**

Karry Burešová

Často potřebujeme načíst nějaký složitý textový vstup: matematický výraz, webovou stránku v HTML, zdroják programu, . . . Ukážeme si, jak texty analyzovat (neboli parsovat), aniž bychom v nich zabloudili: rozdělení na lexikální a syntaktickou vrstvu, železničářský algoritmus na parsování výrazů, popis syntaxe pomocí regulárních výrazů a gramatik. Parsování podle gramatiky: dynamické programování, přístupy LL a LR, packrat parser.

Geometrie a počítače (*“Nerušte mé kruhy! (ani jiné kvadriky)”*) **GEOM**

Jirka Setnička

Základní algoritmy pro řešení geometrických úloh – konvexní obal, dva nejbližší body v rovině, výpočet obsahu nekonvexního mnohoúhelníka, lokalizace bodu, scanline algoritmus a jeho použití, Voroného diagramy a souvislost s persistentními datovými strukturami.

Programovací jazyky a nástroje

Programování v jazyce C **C**

Kuba Maroušek, Filip Štědranský

Jazyk C patří k nejrozšířenějším jazykům, hodí se pro low-level programování i kusy kódu, které mají zejména být rychlé. Představíme si datové typy a běžné programové konstrukce, vysvětlíme si základy práce s ukazateli a také se seznámíme se standardními knihovnamy jazyka C.

Objektově orientované programování nejen v C++ (*“Object-oriented system. If we change it, users object.”*) **OOP**

Jirka Setnička

Objektově orientované programování přináší jiný náhled na návrh řešení problémů. Vysvětlíme, jak se liší objektové a procedurální programování. Co je to objekt a co třída. Základní vlastnosti objektů (dědičnost, zabalení, polymorfismus). Co je to metoda, překrývání metod, virtuální metody (pozdní vazba) a čistě virtuální (abstraktní) metody. Syntaxe a odlišnosti ve třech nejvíce používaných jazycích – C++, C# a Java.

Předpoklady: Znalosti procedurálního programování, například v Pascalu, v Pythonu nebo v C.

Procesy a vlákna * (*“Koupil jsem dalších 15 procesorů, proč je to stále stejně pomalé?”*) **THREAD**

Jirka Setnička

Jak vypadá víceprocesorové či vícejádrové PCčko a co to znamená pro programátora. Procesy, vlákna a úskalí komunikace mezi nimi. Jak se snese n kohoutů na jednom smetišti? Synchronizační primitiva: mutexy, semaforey, podmínkové proměnné. Spinlocky, deadlocky a livelocky. Jde to i bez synchronizace: atomické operace, transakční paměť. Které jazyky nám pomáhají a které spíš škodí. Kdy je lepší vlákna použít a kdy ne.

Předpoklady: Trochu představy o hardwaru

Perl (*“Jak Pejsek a Kočička vymýšleli programovací jazyk”*) **PERL**

Jirka Setnička, Kuba Maroušek, Pali Rohár

Jednoho dne se Larry Wall rozhodl, že nasype do jednoho velkého kotle spousty programovacích jazyků a unixových utilit, za stálého míchání povaří, posléze přecedí, přikoření a implementuje. Tak vznikl Perl, jazyk původně určený hlavně na zpracování textu, ovšem jak se ukázalo, též šikovný na spoustu dalších věcí. Asociativní pole, libovolně složité datové struktury za pomoci referencí, balíčky a objekty zdarma a hlavně regulární výrazy zde a všude. Zkratka jazyk, který lze jedinečně milovat nebo nenávidět, nic mezi tím. Co se Perl 5 přiučil od Perlu 6.

Python (*“print "Ffff".decode("rot13")”*) **PYTH**

Filip Štědranský, Dominik Macháček

Jak programovat v Pythonu a jak v něm „nepsat Cčko“. Syntaxe, datové typy, funkce, třídy, . . . Na co si dát pozor, v čem se Python liší od ostatních jazyků a proč je mezi nimi tak oblíbený.

Logické programování (*“Detektivem za 90 minut.”*) **LOGP**

Pali Rohár

Proč psát dlouhé a složité programy, když stačí dostatečně přesně popsat situaci a pak se prostě zeptat? Toť princip logického programování, který si ukážeme na Prologu.

Jazyk SQL (“*SELECT something FROM knowledge LIMIT 90min*”)

SQL

Kuba Maroušek, Karry Burešová

Dotazovací jazyk SQL a jeho aplikace, čili jak se domluvit s relační databází a zeptat se rovnou na to, co chci vědět. Definice tabulek a indexů. Dotazy a jejich skládání a vnořování. Pohledy, funkce a trigger. Rozdíly mezi dialekty SQL.

Dynamický web a PHP (“*Pepičku, napišeš mi é-šopík?*”)

PHP

Kuba Maroušek, Karry Burešová

Základy praktické tvorby dynamického webu. Úvod do jazyka PHP a Javascriptu, čtení dat z odeslaných formulářů, přesměrování, databáze, generování obrázků a další.

Od zdrojáku k programu (“*Před spuštěním program přeložte. Stačí třikrát podělně?*”)

KOMP

Pali Rohár, Filip Štědranský

Mezi programem v Céčku, který jste právě dopsali, a tranzistory uvnitř vašeho procesoru leží obrovské území obývané překladači, linkery, knihovníky, operačními systémy, loadery a jinými bájnými bytostmi. Pojďme zjistit, co jsou zač a co všechno s programem provádějí. Co udělá kompilátor za nás a co musíme naopak udělat my za něj.

Programování v jazyce C# (“*Co se stane, když strčíme Céčko za mřížku?*”)

CIS

Dominik Macháček

C# je moderní objektově orientovaný jazyk, jehož tvůrci se inspirovali přednostmi a úskalími ostatních programovacích jazyků, zejména Javy. Je jednoduchý a crossplatformní (tedy snadno v něm vytvoříte i okýnka, která nepoběží jen na okýnkách). Naučíme se základy a napíšeme i jednoduchý prográmek.

Hardware a operační systémy

Principy počítačů (“*A opravdu uvnitř počítače běhají malí trpaslíci?*”)

HW

Jirka Setnička, Filip Štědranský

Vydáme se do země skřítků, kteří pohánějí počítače. Počítačové architektury od hodinok po superpočítač od Craye, jejich křivolaká historie i současnost. Co je to procesor, jak se programuje a jak se chová. Různé druhy pamětí a jejich cacheování. Jak procesory komunikují s okolím – sběrnice, čipové sady, vstupní a výstupní zařízení. A co když je procesorů několik, nebo třeba pár tisíc? Přednáška bude praktická: pár počítačů při ní rozebereme a možná i nějaký postavíme.

UNIX (“*UNIX gives you enough rope to hang yourself.*”)

UNIX

Jirka Setnička, Filip Štědranský

Unixové operační systémy (zejména Linux) dobývají svět. Jak fungují uvnitř a jaké nabízejí výhody? Unixová filosofie a historie. Proč je systém složený ze spousty malých a jednoduchých kousků stabilnější a bezpečnější? Proč ovládání prostřednictvím textových příkazů je často efektivnější než klikátka? Jaké to je mít svůj systém pod kontrolou a „vidět mu pod ruce“? V čem spočívá moc textových souborů?

Skriptování v shellu (“*man 1 woman ... man 2 woman ... man group*”)

SHELL

Jirka Setnička, Filip Štědranský

Praktičtěji zaměřená přednáška než UNIX, zabývající se hlavně tím, jak efektivně používat příkazovou řádku. Ukážeme si na spoustě příkladů, jak nám může automatizace všedních činností ulehčit život a jak silné nástroje pro ni UNIXový shell (který navzdory svému názvu existuje i pro Windows) svou jednoduchostí a flexibilitou poskytuje. Budeme spojovat spoustu jednoduchých příkazů do mocných celků a s nimi plnit i na první pohled komplexní úlohy, jako třeba automatické stahování a parsování věcí z webu. Některé činnosti vyžadují lidskou nápaditost a vhled. Ty ostatní bychom měli přenechat strojům.

Programování v Linuxu

PLX

Jirka Setnička, Pali Rohár, Filip Štědranský

Jak si program pod Linuxem povídá s operačním systémem, když chce otevřít soubor, přečíst soubor, půjčit trochu paměti a jiná šprtouchlata. Předvedeme si, jaká existují v Linuxu systémová volání. Naučíme se namapovat si soubor rovnou do paměti, posílat a odchytávat signály, uspávat a probouzet proces, plodit děti a další. Pokud zbyde čas, můžeme si napsat démona a klienta a povídat si po síti.

Předpoklady: Schopnost přečíst a napsat jednoduchý program v C.

Sítě a bezpečnost

Sítě a Internet (“*Sítě nejen na ryby.*”)

NET

Jirka Setnička, Kuba Maroušek

Jak funguje Internet a počítačové sítě vůbec: od elektronů v drátech (fotonů v optických kabelech nebo elektromagnetických vln) přes pakety a jejich routing až k jednotlivým síťovým službám. Adresace, internetworking a dynamický routing. Jak NAT zachránil i zničil Internet a proč se těšíme na IPv6.

Sítě II – protokoly a síťové útoky (“*Jak si přečíst mailly. ... sousedovy mailly.*”)

NET2

Jirka Setnička

Volné navázání na NET. Budeme si povídat o tom, co za data nám po síti běhá a jaké se k tomu používají protokoly – DNS, FTP, HTTP nebo třeba i mailové SMTP a IMAP. Zaměříme se více na ty nejpoužívanější (metody GET a POST v HTTP), nakousneme cacheování a nadlábneme se cookies. A pokud zbude čas, využijeme zranitelnosti některých protokolů a provedeme síťový útok.

Předpoklady: Základní povědomí o počítačových sítích

Jirka Setnička

Kryptologie čili tajuplná nauka o šifrách, jejich konstrukci a hlavně o jejich luštění. Přísně tajné. Šifrovací systémy jako lego: základními kostičkami nám budou symetrické a asymetrické šifry a jednosměrné funkce, stavět z nich budeme kryptografické protokoly na bezpečný přenos, autentikaci, digitální podpisy a třeba i na házení korunou po telefonu. Předvedeme nerozluštitelnou šifru a dokonce to o ní i dokážeme.

Teoretická informatika

Složitější složitost *

SLOZ2

Karel Tesař

Trochu hlouběji o složitosti. Přesná definice výpočetního modelu a velikosti vstupu. Složitost v nejlepším, nejhorším a průměrném případě; amortizovaná analýza. Jak dokázat, že úlohu nejde řešit rychleji, aneb dolní odhady. Porovnávání problémů pomocí redukcí, problémy NP-úplné a ještě těžší.

Předpoklady: SLOZ

Evoluční algoritmy * (*“Já to dělat nebudu, ať to za mě udělají mravenci!”*)

EVA

Karel Tesař, Karry Burešová

Evoluční algoritmy se inspiroují strukturami chování v přírodě a na jejich základě pak (optimalizačně) hledají řešení těžkých problémů. Na přednášce určitě zazní genetický algoritmus, zmíníme jeho algoritmy a když zbyde čas tak si obecněji popovídáme o algoritmech pohybujících se ve velkých prostorech řešení.

Strojové učení * (*“Umí počítače přemýšlet?”*)

MACHINE

Karel Tesař, Karry Burešová

Úlohy, kterými se zabývá strojové učení. Co je to učení s učitelem a bez učitele? Ukážeme si základní postupy, jak počítač můžeme naučit rozpoznávat věci, které sami rozpoznat neumíme. Umí Facebook nebo Google poznat, co máme rádi, jakou máme náladu, nebo jestli jsme těhotní?

Modely počítačů (*“Nač Pentium? Máme Turingovy stroje!”*)

MODEL

Jirka Setnička

V HW se dozvíte, jak fungují „opravdové“ počítače, zde pro změnu na čem počítají teoretici. Všechny počítače jsou si rovny, jen některé jsou si rovnější. Turingův stroj obyčejný, vícepáskový, nedeterministický a univerzální. Random Access Machine (RAM) a Pointer Machine. Až nám začne být smutno, pořídíme si klidně N^2 procesorů a spráhneme je do paralelního počítače (PRAM). Rychlé paralelní slévání a třídění. Pokud zbude čas, ukážeme si buněčné a grafové automaty, nebo třeba dlaždičky v koupelně.

Jazyky, gramatiky a automaty *

AUTO

Jirka Setnička, Karry Burešová, Dominik Macháček

O jazycích přirozených, počítačových a matematických, jejich popisu a rozpoznávání. Začneme těmi nejjednoduššími: regulární jazyky a výrazy, konečné deterministické a nedeterministické automaty. Pak budeme stoupat po příčkách Chomského hierarchie, kam až to půjde. Jak výpočetně silný je třeba takový automat na kafe?

Matematické přednášky

Grafy bez algoritmů

GRAFY

Karry Burešová

Teorie grafů trochu teoretičtěji. Různé druhy grafů a jejich vlastnosti. Stromy a lesy. Kreslení grafů jedním tahem. Princip sudosti a skóre grafu. Jaké speciální vlastnosti mají rovinné grafy a jak je lze obarvit šesti nebo možná i pěti barvami. Jak poznat, že dva grafy (ne)jsou isomorfní. Mosty, artikulace a ušaté lemma. Párování, střídavé cesty a Hallova věta.

Úvod do teorie čísel (*“Po malém fermetu mívám čínský zbytkáč.”*)

NUT

Karry Burešová

Co a k čemu je teorie čísel. Počítání v kongruenci, Euklidův algoritmus a jeho použití. Konečná tělesa a Malá Fermatova věta. Prvočísla a Eratosthenovo síto. Čínská zbytková věta a její algoritmická verze. Jak si odvodit kritéria dělitelnosti.

Kombinatorika (*“Nemám rád faktoriály. Faktoriály nemám rád. Rád nemám faktoriály. . . ”*)

KOMB

Při navrhování algoritmů a počítání jejich složitosti narazíme na celou řádku zajímavých a ne úplně triviálních kombinatorických problémů, a tak se naučíme, jak na ně. Základní triky s faktoriály a kombinačními čísly, sčítání konečných a občas i nekonečných řad, rekurentní rovnice a princip inkluze a exkluze.

Rozšiřující přednášky

Mezi rozšiřujícími přednáškami se dají nalézt různé specifictější obory a zájmy, jakožto i těžší přednášky navazující na předchozí díly ze základních přednášek. Mezi nabízenými přednáškami si tak můžete vybrat obor svého zájmu a tomu se dále věnovat.

Algoritmy a datové struktury

Datové struktury pro šílence ** (*“A tohle fakt jde naprogramovat? Tomu nevěřím!”*)

DS3

Karel Tesař

Jak postavit datovou strukturu, která si pamatuje svou vlastní historii? Jak z libovolné statické datové struktury udělat dynamickou? Lze navrhnout vyhledávací strom se složitostí $\mathcal{O}(\log \log n)$? Tyto a ještě další triky budou obsaženy v této přednášce.

Nejkratší a jiné cesty * (*“Všechny cesty vedou do Horní Dolní, jen některé přes Řím.”*)

CESTY

Jirka Setnička

O problému hledání cest v grafech trochu podrobněji. Obecné relaxační schéma, Bellmanův-Fordův a Dijkstrův algoritmus a jejich zrychlení pomocí různých datových struktur. Potenciálová redukce a heuristiky (třeba A^*), zaokrouhlování délek hran. Souvislosti s násobením matic: transitivní uzávěr, Seidelův algoritmus, Kleeneho algoritmus a regulární výrazy.

Sufixový strom a sufixové pole ** (*“Umíš sufixový strom? Tak to už s řetězci umíš úplně všechno!”*)

SUFFIX

Karel Tesař, Filip Štědranský

Pomocí sufixového stromu lze většinu řetězcových problémů vyřešit v lineárním čase. Na přednášce si ukážeme, jak sufixový strom vypadá, jak se pomocí něj řeší problémy, jak souvisí se sufixovým polem a pomalu se budeme přibližovat k jeho konstrukci.

Metody řešení NP-úplných problémů ** (*“I NP-úplné problémy jdou řešit.”*)

SOLVENP

Karel Tesař

NP-úplný problém je problém, pro který není znám žádný algoritmus řešící optimálně všechny instance v polynomiálním čase. To ale neznamená, že takový problém vůbec vyřešit neumíme. Podíváme se na příklady ”efektivních” exponenciálních algoritmů a další způsoby nakládání s NP-úplnými problémy.

Algoritmy na stromech *

STROM

Karel Tesař

V případě stromů se mnohé těžké problémy stávají jednoduchými a mají svá specifická řešení. Ukážeme si použití myšlenky dynamického programování, vyřešíme úlohu nejbližšího společného předchůdce a předvedeme dekompozici stromu a následně postavení datové struktury nad ní.

Kompresce dat (*“Jnm idln kpln j nstlčtn.”*)

PRESS

Jirka Setnička

Přehled základních kompresních algoritmů: triviální algoritmy (RLE), statistické metody (Huffmanovo a aritmetické kódování), slovníková komprese (LZ77, LZ78, LZW), Burrowsova-Wheelerova transformace (BZIP). Pokud zbude čas, tak i něco o ztrátové kompresi obrázků a zvuku (prediktory, wavelets, JPEG, MPEG, fraktály).

Magické algoritmy * (*“Pokročilá magie není rozlišitelná od technologie.”*)

MAGIC

Filip Štědranský

O algoritmech značně magických a nečekaných. Jak násobit n -ciferná čísla rychleji než v kvadratickém čase. Kouzlo na slévání seřazených posloupností v konstantním prostoru. Isomorfismus stromů pomocí přihrádkového třídění. Bitové kejklřství. Hledání největší díry.

Programovací jazyky

Černá magie v C++ * (*“Je dobré znát, co umí atomová bomba (a její datový typ), abychom ji nechtěli použít.”*)

CPP

Filip Štědranský

Pokročilejší prvky C++ (šablony, přetěžování funkcí a operátorů, preprocesorové hacky, ...) a možnosti jejich (po|z)užití. Jak s nimi vytvořit věci magické (smart pointers, lambda funkce), ale i na první pohled docela obyčejné, které si bohužel jinak pořídit nejde. Nahlédneme do vnitřností slavné knihovny Boost. Proč je třeba preprocesorem vygenerovat 512 šablonovaných typů jen, abychom mohli rozumně předávat pointery na metody? Jaká akrobacie je nutná pro obyčejnou statickou inicializaci seznamu? A mnohá další překvapení. Přednáška vás naučí psát v C++ složité věci a přesvědčí, že v něm nechcete psát ani jednoduché.

Předpoklady: základní znalost C++, staticky alokovaný kyblík

Perl 6 (*“Slečno, mohu vám ukázat svou sbírku operátorů?”*)

P6

Filip Štědranský

Je to Perl, a přitom to Perl není. Co je to? Aneb jak to dopadne, když se pokusíme navrhnout programovací jazyk budoucnosti a inspirovat se přitom filosofií Perlu. Typový systém, pokud zrovna chcete. Objekty, třídy a metatřídy. Periodická soustava (meta)operátorů. Definování jazyka v sobě samém. A co se to stalo s regulárními výrazy? Jak vypadají implementace P6 a kdy je prozatím lepší programovat na papíře. Praktické cvičení ve stavbě vzdušných zámků a bydlení v nich.

Filip Štědranský, Dominik Macháček

Povídání o méně zmiňovaných částech Pythonu. New-style classes, dekorátory, metaklasy, generátory, funkcionální styl programování v Pythonu. Jak napsat quicksort jako lambda funkci. Představení zajímavých modulů nejen ze standardní knihovny.

Předpoklady: PYTH

Programovací nástroje a techniky

Git a jiné systémy pro správu verzí (“U svatýho tučňáka, kdo sem napsal tohle? Ono to tvrdí, že JÁ?!”)

GIT

Jirka Setnička, Pali Rohár, Karry Burešová

Jak vyvíjet program delší dobu a nezbláznit se u toho. Různé systémy pro správu verzí od diff/patch přes CVS a SVN až ke Gitu. Jak Git funguje: stromy, commity, větve, tagy. Merge mezi větvemi nebo mezi různými počítači. Kouzelnické triky: hledáme bugy púlením historie, přepisujeme dějiny. Jak se liší správa zdrojáků v projektech o jednom, deseti a tisíci programátorech. Udržujeme patche k cizímu programu aneb quilt a StGit.

Make (“make love ... don't know how to make love”)

MAKE

Pali Rohár

Hodil by se otrok, který by překládal jednotlivé soubory. Základní syntaxe takového otroka, jak napsat jednoduchý Makefile, který řeší překlad Céčkového programu, automatické řešení závislostí. Jak to udělat, aby výsledek neměl několik tisíc řádek. Proč by se hodilo, aby tu bylo něco lepšího. A proč automake, cmake a qmake nepomáhají.

Gdb a jiné ladicí nástroje * (“Jak se ladí kytara, jak křišťálová koule a jak program (řazeno dle obtížnosti)”)

GDB

Pali Rohár

Kdo píše programy, které vždy hned fungují, ať se přihlásí. A kdo ne, ať se přihlásí na tuto přednášku. Ukážeme si několik nástrojů, jak si pomoci z nejhoršího. Mezi nimi třeba gdb, řádkový debugger (odvšivovač), strace, nebo valgrind. Kdy je použít a kdy se více hodí printf. Proč assert je tak užitečná věc.

Textový editor Vim (“Víš, jaký je nejlepší textový editor? Vim.”)

VIM

Filip Štědranský, Karry Burešová

Odložme na chvíli své myši a pojďme si vyzkoušet textový editor, který umí poslouchat na slovo. Pravda, budeme se ta slova muset chvíli učit, ale výsledek bude proklatě efektivní. Základní příkazy, práce s regulárními výrazy, makra, kouzla. Vimovité ovládání jiných programů, třeba webového prohlížeče.

Jak se nestat vepřem (“/* You are not expected to understand this */”)

STYLE

Karry Burešová

Tvrdí se, že čist kód je mnohdy těžší, než ho psát – dokonce i po sobě, stačí krátká doba. Je několik obecně uznávaných pravidel, jak kód psát a jak ne, aby byl hezký a dobře čitelný. Od základních (rozumná pojmenovací konvence, systematické odsazování), až po to, kdy opravdu použít goto a jak napsat užitečný komentář nebo dokumentaci. A kdy se vyplatí se na všechna tato pravidla vybodnout.

Hardware a operační systémy

Cache-oblivious algoritmy (“Kešuješ, kešuje, kešujeme”)

CACHE

Jirka Setnička

Dnešní procesory mají několik úrovní vyrovnávacích pamětí (cache), což způsobuje, že ačkoliv si jsou všechny části paměti rovny, některé si jsou rovnější. Jak taková cache funguje? Jak se procesor rozhodne, co si v ní zapamatuje a co vyhodí? Jak toho můžeme využívat při programování, aby naše programy běžely rychleji? Předvedeme kousek teorie i několik praktických ukázek s poněkud překvapivým chováním.

Předpoklady: Kešu oříšky

Linuxové jádro a jak se v něm vyznat (“Jak pořádně otestovat fsck?”)

KERN

Pali Rohár

Co ten kernel vlastně je, čím se liší programování v kernelu od normálního kódu, jak sobě vlastní kernel postavit a jak v něm něco opravit. Kde najít nejnovější zdrojáky a kde najít pomoc, až se něco pokazí.

Sítě a bezpečnost

E-mail (*“Drahoušek zákazník.”*)

EMAIL

Kuba Maroušek, Pali Rohár

Co se stane s e-mailem, když jej odešlete? Kudy chodí a kudy jej čerti nesou? Jaké máte záruky, že přijde; proč občas přijde pozdě nebo vůbec. Problém formátů a kódování, chyby webových i jiných klientů. Protokoly SMTP, POP, IMAP a co se stane, když do nich přimícháme SSL/TLS. E-mailová bezpečnost, SPAM, viry, phishing, BFU a kde koupit levnou viagru. Nakonec státní datové schránky a proč je to zlý ošklivý nepěkná věc. A jak se správně podepsat. A jako bonus sociální síť, ve které je každý a ani o tom neví.

Aplikace kryptografie * (*“6140 a184 c9a6 41f1 de99 e733 354a f451”*)

CRYPT2

Filip Štědranský

Pokročilejší a občas nečekané aplikace základních kryptografických primitiv. Jak přesvědčit server, že známe heslo, aniž bychom mu ho poslali? Jak zajistit, aby útočník nemohl dešifrovat komunikaci, ani když dodatečně získá soukromý klíč? Jak funguje BitCoin (decentralizovaná digitální měna) či Tor (protokol znemožňující komukoli po cestě vědět, kdo s kým komunikuje)? Malý bonus: nečekané útoky na kryptografické systémy. Jak zjistit heslo na základě doby odpovědi serveru? Jak otevřít libovolné auto krabičkou za 30 dolarů? A kdy se při hackování hodí tekutý dusík?

Předpoklady: Základní povědomí o šifrování (CRYPT) a víra v existenci náhodných čísel

Grafika a typografie

Počítačová grafika (*“Namaluj mi beránka. . .”*)

GFX

Jirka Setnička

Kreslení a zpracování obrazu na počítači. Co vše obnáší vykreslení obyčejné čáry, aby to bylo rychlé a pěkně vypadalo. A co teprve, když ta čára zatáčí! Vyplňování n -úhelníků a křivkou ohraničených oblastí, flood fill. Také maticové filtry pro zpracování fotek (zaostření, rozmazání), anti-aliasing a dithering. Pokud se stihne, tak navíc základy 3D vykreslování.

MetaFont, MetaPost (*“Teď ten obrázek takhle zkrouším a pak ho přeložím.”*)

MF

Jirka Setnička

Lehké nakousnutí jazyka, ve kterém můžete opravdu kreslit planimetrické obrázky, ale i třeba písma nebo piktogramy do zadání a řešení KSP. Jak vypadají CM fonty (ty, které používá \TeX) a jak se autorovi povedlo, že se z jediného „obrázku“ dá vygenerovat tlusté, tenké, rovné, skloněné, šišaté písmenko.

Typografie (*“What You See Is all What You’ve Got!?”*)

TYPO

Karry Burešová

Jak na počítači text nejen napsat, ale také vysázet tak, aby pěkně vypadal a aby (což je důležitější) se i příjemně četl. Jak se sází pohádka, jak báseň a jak vzorové řešení KSP plné komplikovaných vzorců. Jak jde dohromady staleté umění typografické a moderní technika. Přineste knihy i letáky, zkritizujeme sazeče, co se do nich vejde.

\TeX (*“No pages of output. Ask a \TeX nician.”*)

TEX

Jirka Setnička, Kuba Maroušek

Donald E. Knuth napsal \TeX před desítkami let proto, že mu nikdo nebyl schopen vysázet matematický text podle jeho požadavků. Od té doby se hojně používá pro sazbu nejrůznějších publikací. V této spíše praktické přednášce si ukážeme použití \TeX u od hladké sazby knihy až po zbesilosti hraničící s programováním. Pozornost věnujeme i zdrojům informací a rozdílům mezi různými dialekty \TeX u.

Aplikace informatiky

Počítačová lingvistika (*“Jsou bramborové knedlíky plněné bramborami?”*)

CMPLING

Karry Burešová

Zejména motivační přednáška o počítačové lingvistice a počítačovém zpracování přirozeného jazyka. Podíváme se na vlastnosti přirozených jazyků a zaměříme se na to, jak moc komplikují jejich počítačové zpracování. Pojmenujeme odlišnosti mezi kontrolou pravopisu, automatickým překladem a konverzací s uživatelem a ukážeme si, co se zatím umí používat.

Testování uživatelského rozhraní (*“Vždyť to tlačítko je tak evidentní!”*)

TUR

Karry Burešová

Obvykle tvoříme programy (nebo třeba webové stránky) s cílem, aby je používali i další lidé. K tomu je ovšem vhodné, aby se i ostatním dobře používali. Jak něco takového měřit a testovat? Kognitivní průchod, heuristická evaluace i testování s lidmi. Jak pracovat s výsledky testů a proč nevádí, že Vim by v některých testech rozhodně neuspěl.

Mobilní zařízení (*“Co je malé, to je hezké, a když ne, tak toho aspoň není moc.”*)

MOBI

Jirka Setnička

Jak se liší PDA, MDA, Smartphone a obyčejný mobilní telefon. Jak vlastně takový mobilní telefon funguje a jak vypadá struktura základnových stanic dovolující, aby fungovat mohl. Spíše hardwarový přehled toho, co vlastně v dnešním telefonu je a co to umí – dotykové technologie, bezdrátové sítě, vlastnosti různých baterií, metody šetření elektřinou. Diskuze o třech (čtyřech) hlavních platformách dneška a ukázka konkrétního využití, aneb s telefonem se dá mnohem více, než jen telefonovat.

Ako si postaví bezpečný telefon (*“To naozaj neexistuje telefón, ktorý by svojho majiteľa neodpočúval?”*)

NEO900

Pali Rohár

Mobilný telefón dnes bežne používame na zadávanie platobných príkazov či okamžité zdieľanie fotiek cez internet. Je to ale bezpečné? Pozrieme sa aké súčasti mobilné telefóny obsahujú, ako sú jednotlivé HW/SW komponenty zabezpečené a čo nám môže spraviť útočník, výrobca telefónu alebo mobilný operátor. Ukážeme si koncept bezpečnejšieho mobilného telefónu, konkrétne projekt Neo900.

Matematické přednášky

Lineární programování *

LINPRG

Karel Tesař

Jakýkoliv problém, který lze popsat soustavou lineárních nerovnic lze vyřešit v polynomiálním čase. A to metodou lineárního programování. Povíme si základy k tomu, jak lineární program vypadá, jak se vyřeší a jak různé úlohy dají popsat pomocí lineárního programu. Přednáška se může buď více zaměřit na samotnou teorii lineárního programování, a nebo na jeho aplikaci v různých algoritmických úlohách.

Lineární algebra

LA

Kuba Maroušek

Lineární algebra původně vznikla jako elegantní prostředek k popisování geometrie lineárních útvarů (bodů, přímek, rovin, ...) v libovolněrozměrném prostoru, ale ukázalo se, že její kouzlo dosahuje daleko dál. Vektorové prostory, lineární (ne)závislost, báze, lineární zobrazení a matice, determinanty, tenzory. Konečné projektivní roviny.

Komplexní a komplexnější čísla (*“ $1 = \sqrt{1} = \sqrt{(-1)(-1)} = \sqrt{-1}\sqrt{-1} = i \cdot i = i^2 = -1$. Huh?”*)

CPLX

Janka Bátoryová, Martin Šerý

Jak se nám matematika změní, když připustíme, že se záporná čísla také dají odmocňovat? Čísla imaginární a komplexní a jejich různé podoby. Součtové vzorce pro sin a cos dostaneme téměř zdarma. K čemu se hodí v matematice a k čemu ve fyzice. Proč se zastavit u dvou složek aneb kvaterniony, oktoniony a Cliffordovy algebry. Remember, life is complex.

Úvod do klasické analýzy * (*“Dej pokoj a už mě konečně zinfinitizuj”*)

KA

Kuba Maroušek

Limita, derivace, derivace⁻¹, integrál. Z pohledu formální matematiky velmi podobné si jsou. My si ukážeme, jak jednoduše lze tyto pojmy pochopit a že vyzrát na ně není vůbec těžké.

Deskriptivní geometrie (*“Jak splácnout tři rozměry do dvou”*)

DG

Janka Bátoryová

Jemný úvod do Mongeova promítání a axonometrie. Jak nakreslit krychli aby vypadala „jako živá“, jak narýsovat na list papíru dvě navzájem kolmé roviny a poznat, kde se protínají a spousta dalších zajímavých konstrukcí.

Půlnoční přednášky

Aneb přednášky přednášené (nejen) o půlnoci na různá zajímavá témata nejen o informatice. Pokud nějaká z nich nebude oficiálně vypsaná, je možné si konkrétního organizátora ve volné chvíli chytit a přesvědčit ho k přednášení.

Lingvistika (*“Přísudek je v této větě podmět.”*)

LING

Karry Burešová

Převážně nevážené a mírně nepřed-vídatelné po-vídání o jazyku i jazyce. Základní jazykové rodiny a jejich podobnosti i odlišnosti. Co má společného čínština s angličtinou a co nikoliv. Proč jeden jazyk potřebuje 15 pádů, zatímco jiný se bez nich obejde úplně. Jak se jazyky vyvíjejí a jak se navzájem ovlivňují. Kde se berou jazyková pravidla. Kde se vzalo písmo a proč se mluvený a psaný jazyk tolik liší. Jak se na jazyk dívá matematik a jak se na matematiku dívají lingvisté.

MFF UK aneb co obnáší matfyzákem býti (*“Maminko, ptá se tatínka, kdy už budu matfyzákem?”*)

MFF

Nezávazné povídání o Matfyzu a základním matfyzáckém folkloru. Určitě si přečteme matfyzáky sepsané Úvod do matfyzáka a zazpíváme pár matfyzáckých písní. Zbytek už bude záležet na tom, co budete chtít slyšet.

Jak uspět v programátorských soutěžích

PSOUT

Karel Tesař

Co mám dělat, abych uspěl v soutěžích? Jak se připravovat? Na co dát důraz? Co si připravit předem a čeho si všimnout při soutěži? Jak odhadnout náročnost a které úlohy si pro řešení vybrat jako první? Je úspěch otázka talentu nebo tvrdého tréninku?

Organizování (*“Ten dělá to a ten zas tohle aneb co obnáší organizátorem býti”*)

ORG

Jirka Setnička

Volné povídání o tom, co se všechno skrývá za organizováním různých seminářů a podobných akcí, primárně pak KSPčka. Jaká práce, jaké radosti a jaké starosti s sebou organizování nese, co se přitom člověk může naučit a také pár cenných rad do života. Jak se z toho nezbláznit a pár bláznivých příhod k tomu.

Orientace

ORI

Jirka Setnička

Jak ze neztratit v terénu a jak se neztratit na moři. Vývoj umění navigace. K čemu je důležité slunce a hvězdy, ale proč mořeplavcům nestačí, alespoň dokud neobjevíme hodinky. Použití mapy, busoly a GPSky. Orientace bez pomůcek a použití Ariadniny nitě. Bleskový úvod do sférické astronomie a časoměry čili jak (ne)postavit sluneční a třeba i měsíční hodiny. Jak reprezentovat mapu v počítači a jak raději ne. Jak zapisovat polohu místa na Zemi (přestože Země má tvar podivně nakousnuté hrušky) a kolika způsoby to jde. Různé druhy map a jejich (z)kreslení. Jak se neztratit v kartografii. Praktické cvičení v terénu.

Základy první pomoci (*“Jak někomu zachránit život a jak málo k tomu stačí”*)

ZDRAV

Jirka Setnička, Karry Burešová

Pobavíme se o základech první pomoci. Jak správně vyhodnotit situaci a kdy je potřeba volat pomoc? Jak se postarat o člověka v bezvědomí, jak kontrolovat životní funkce a jak člověka stabilizovat do příjezdu pomoci? Ukážeme si, jak málo stačí k záchraně života a naučíme se nebát se první pomoci. A také, že naše bezpečí je v každé situaci na prvním místě.

Předmětové olympiády od A do Z

SOUT

Karry Burešová

České předmětové olympiády z pohledu soutěžícího i nezávislého pozorovatele. Jak se dostat do celostátního kola, jak (možná) dojít až do mezinárodní olympiády a která cesta vede zaručeně do pekel. Příspěvek ze strany korespondenčních seminářů, aneb zapomeňte školní znalosti, ty vám nepomůžou. Nečekejte univerzální rady, neb žádné takové neexistují, spíše vyprávění o cestě obyčejného smrtelníka olympiádním molochem.

Auto z pohledu technika (*“Co mi to vrčí pod kapotou a proč bliká ta kontrolka?”*)

CAR

Jirka Setnička

Nahlédneme do tajů starších i novějších aut. Podle zájmu se můžeme pobavit o tom, jaký je rozdíl mezi benzínovým a naftovým motorem, či proč se auta staví zrovna tak, jak se staví. Na praktické ukázce probereme (a trochu rozebereme) auto a co nejvíce si ukážeme – z pohledu běžné údržby i jednoduchých oprav. Určeno pro každého, koho čeká autoškola, nebo ho jen baví mechanika.

Technika ve sci-fi (*“Scotty, beam me up”*)

SCIFI

Jirka Setnička

Technické objevy v různých sci-fi (Star Trek/Gate/Wars i dalších) a pohled na ně z perspektivy dnešních fyzikálních znalostí. Proč se v Hollywoodských filmech ozývá ve vesmíru zvuk laserů, i když je tam vakuum, a je možné cestovat rychleji než světlo? Také možná zabrousíme do některých filozofických otázek – primární směrnice o nevměšování ve Star Treku a jiné.

Biologická evoluce (*“Ale vždyť je to jen teorie!”*)

EVOL

Filip Štědranský

Evoluční teorie je asi jedním z největších převratů v lidském náhledu na svět. Jak se na takovou věc vůbec přišlo, proč to tak dlouho trvalo a proč si myslíme, že je to pravda? Jak si to vše alespoň trochu představit? Mnoho nečekaných překvapení, informatické a fyzikální souvislosti. Proč jsou stromy vysoké, jak souvisí pohlavní rozmnožování s počítačovou bezpečností, co má evoluce společného s Windows a co s halting problémem?

Předpoklady: Základní přednáška nevyžadující předchozí biologické znalosti.

Shell pro shíllence (“mkfifo p;nc -lp80<p|sed -re "s/GET /tac</s/ .*/;echo;echo& 200 OK;/q"|sh|tac>p”) **SS**
Filip Štědranský

Co vše se dá napsat za pomoci unixového shellu a spřízněných utilit. Webové aplikace, šablonovací systém, síťová komunikace, OOP, ... Proč to obvykle není dobrý nápad, ale občas taky ano.

Filosofie programovacích jazyků (“*Your programming language sucks!*”) **YPLS**

Filip Štědranský

Společné filosofování nad programovacími jazyky a zejména nad tím, proč jsou všechny na draka, jen některé víc než jiné. Proč lze mít všechno na světě, jenže ne najednou. Proč je někdy lepší nechat programátora, aby si sedl na své vlastní vidle. Proč jsou některé vlastnosti jazyků na první pohled geniální, zatímco jiné (až) na ten druhý. Proč je důležitá hustota kódu a Huffmanův princip. Proč je někdy důležitější evoluce než revoluce.

Svoboda a soukromí v digitálním světě **DRIGHTS**

Filip Štědranský

Kdo, proč a jak se nás o ně snaží připravit? Co o vás ví Google, Facebook, Apple či NSA a jakou nad vámi mají moc? Kdo může z vašeho telefonu vzdáleně číst a mazat data? Čím hrozí „kyberzákony“ a mezinárodní dohody? DMCA, SOPA, ACTA, TT(I)P a další. Bude Hollywood rozhodovat o tom, co (ne)smíte vidět na internetu a Microsoft o tom, jaký software (ne)smíte používat? Proč je často protizákonné upozornit uživatele na bezpečnostní chyby? A kdy váš to vše může stát život?

Hudební nauka pro smrtelníky (“*Stupnice se skládá ze sedmi trpaslíků, nebo dvanácti měsíčků?*”) **MUSIC**

Kuba Maroušek

Povídání o hudbě, hudebních stupnicích a jiných věcech spojených s hudbou a její reprodukcí (primárně kytarovou).

Co je to spravedlnost? (“*Ale to není fér!*”) **JUSTICE**

Dominik Macháček

Co je to spravedlnost? Co je správné? Je na světě něco spravedlivé? Existuje objektivní spravedlnost? Kdo za ni ručí a kde se na světě vzala? Nebo je to jen prospěch silnějšího a lidský výmysl? Pokusíme se vést rozhovor, jehož účelem bude najít pravdu. Můžeme použít třeba sokratovskou metodu.

Abecední seznam přednášek

LYK Stručný úvod do základů teorie vlkodlaků.. 1

Základní přednášky

SLOZ	Algoritmy a jejich složitost.....1	KOMP	Od zdrojáku k programu.....3
DS2	Datové struktury pro pokročilé 1	PARSE	Parsing čili analýza textu 2
DS1	Datové struktury pro začátečníky 1	PERL	Perl 2
DYNP	Dynamické programování 2	HW	Principy počítačů.....3
PHP	Dynamický web a PHP 3	THREAD	Procesy a vlákna 2
EVA	Evoluční algoritmy 4	C	Programování v jazyce C 2
GEOM	Geometrie a počítače 2	CIS	Programování v jazyce C#.....3
GA	Grafy & algoritmy 1	PLX	Programování v Linuxu 3
GRAFY	Grafy bez algoritmů 4	DFS	Prohledávání do hloubky 1
REGEX	Hledání v textu 2	PYTH	Python 2
ITREE	Intervalové stromy 1	NET	Sítě a Internet 3
SQL	Jazyk SQL 3	NET2	Sítě II – protokoly a síťové útoky 3
AUTO	Jazyky, gramatiky a automaty 4	SHELL	Skriptování v shellu 3
KOMB	Kombinatorika 4	SLOZ2	Složitější složitost.....4
CRYPT	Kryptologie 4	MACHINE	Strojové učení 4
LOGP	Logické programování.....2	TOKY	Toky v sítích 1
MODEL	Modely počítačů 4	UNIX	UNIX 3
OOP	Objektově orientované programování nejen v C++ 2	NUT	Úvod do teorie čísel.....4
		ZALG	Základní algoritmy 1

Rozšiřující přednášky

NEO900	Ako si postaviť bezpečný telefón	8	MAKE	Make	6
STROM	Algoritmy na stromech	5	MF	MetaFont, MetaPost	7
CRYPT2	Aplikace kryptografie	7	SOLVENP	Metody řešení NP-úplných problémů	5
CACHE	Cache-oblivious algoritmy	6	MOBI	Mobilní zařízení	8
CPP	Černá magie v C++	5	CESTY	Nejkratší a jiné cesty	5
DS3	Datové struktury pro šílence	5	P6	Perl 6	5
DG	Deskriptivní geometrie	8	GFX	Počítačová grafika	7
EMAIL	E-mail	7	CMPLING	Počítačová lingvistika	8
GDB	Gdb a jiné ladicí nástroje	6	PYTH2	Pokročilé povídání o Pythonu	6
GIT	Git a jiné systémy pro správu verzí	6	SUFIX	Sufixový strom a sufixové pole	5
STYLE	Jak se nestat vepřem	6	TUR	Testování uživatelského rozhraní	8
CPLX	Komplexní a komplexnější čísla	8	TEX	TeX	7
PRESS	Komprese dat	5	VIM	Textový editor Vim	6
LA	Lineární algebra	8	TYPO	Typografie	7
LINPRG	Lineární programování	8	KA	Úvod do klasické analýzy	8
KERN	Linuxové jádro a jak se v něm vyznat	6			
MAGIC	Magické algoritmy	5			

Půlnoční přednášky

CAR	Auto z pohledu technika	9	ORG	Organizování	9
EVOL	Biologická evoluce	9	ORI	Orientace	9
JUSTICE	Co je to spravedlnost?	9	SOUT	Předmětové olympiády od A do Z	9
YPLS	Filosofie programovacích jazyků	9	SS	Shell pro šílence	9
MUSIC	Hudební nauka pro smrtelníky	9	DRIGHTS	Svoboda a soukromí v digitálním světě	9
PSOUT	Jak uspět v programátorských soutěžích ...	9	SCIFI	Technika ve sci-fi	9
LING	Lingvištika	9	ZDRAV	Základy první pomoci	9
MFF	MFF UK aneb co obnáší matfyzákem být .	9			