Podzimní soustředění KSP 2014 – 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**
Martin Mareš, Karolína Burešová, Filip Štědronský, Lucie Mohelníková
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**
Martin Mareš, Michal Vaner, Jirka Setnička, Jenda Hadrava
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**
Michal Vaner, Martin Mareš
Důmyslnější varianty vyhledávacích stromů: splay stromy, BB-α 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, Karolína Burešová
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**
Karolína Burešová, Filip Štědronský
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**
Martin Mareš, Jirka Setnička, Filip Štědronský, Karolína Burešová
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**
Martin Mareš, Jirka Setnička, Karolína 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**
Martin Mareš, 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**
Michal Vaner, Filip Štědronský, Jan Kubálek, Karolína Burešová
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 knihovnami jazyka C.

Objektově orientované programování nejen v C++ *("Object-oriented system. If we change it, users object.")* **OBJ**
Michal Vaner
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 v jazycích C++, `C#`, Java, Object Pascal, či úplně jiné přístupy v jazycích jako Obj C, Perl, Erlang, ...
Předpoklady: Znalosti procedurálního programování, například v Pascalu, v Pythonu nebo v C.

Java **JAVA**
Karolína Burešová
Základy syntaxe, základní typy. Třídy, dědičnost, interface. Práce s objekty, s poli a s řetězci. Povídání o alokaci paměti a garbage collectoru. Zpracování výjimek. Jak na vlákna a jejich synchronizaci.

Generika * *("Má C typovou kontrolu? Ano, ale jen občas.")* **TEMPL**
Michal Vaner
Co je to generická struktura, jak v C napsat spojový seznam, spojovou mřížku, kde se do toho hodí `void *`, dědičnost (ano, v C) a preprocesor. Šablony v C++, aneb neexistuje věc, která by nešla napsat, jen existuje spousta, které se nevyplatí. Jak to řeší jiné jazyky (Java, Haskell, Perl) a jakou za to platíte cenu.
Předpoklady: Přibližná znalost C, C++ a možná dalších, kyblík

Procesy a vlákna * *("Koupil jsem dalších 15 procesorů, proč je to stále stejně pomalé?")* **THREAD**
Michal Vaner, Martin Mareš
Trochu více praktická přednáška o paralelním programování, než PARAL. 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, semafory, 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**
Martin Mareš, Michal Vaner, Jan Kubálek, Lucie Mohelníková, Jirka Setnička, Karolína Burešová
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. Zkrátka jazyk, který lze jedině milovat nebo nenávidět, nic mezi tím. Co se Perl 5 přiučil od Perlu 6.

Python *(“`print "Ffff".decode("rot13")`”)* **PYTH**
Michal Vaner, Jirka Setnička, Filip Štědronský, Karolína Burešová
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**
Jirka Setnička
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.

Haskell *(“V téhle proměnné je uložen okolní svět.”)* **HASK**
Michal Vaner, Jirka Setnička, Jan Kubálek, Jenda Hadrava, Karolína Burešová
Základní kurz Haskellu – moderního funkcionálního jazyka. Ukážeme si syntaxi, vysvětlíme typovou kontrolu a typový systém. Přičichneme k třídám, zrušíme výjimky a zavedeme zcela bezpečná vlákna. Řekneme si, proč v Haskellu nejde komunikovat s okolním světem a proč nám pomůže si okolní svět uložit do proměnné. A že vlastně v haskellu žádné proměnné nejsou, jen visačky na datech.
Předpoklady: Sklony k algebraickému chápání vesmíru, odvahu tváří v tvář své vlastní tváři a rekurzi.

Jazyk SQL *(“SELECT something FROM knowledge LIMIT 90min”)* **SQL**
Jirka Setnička, Karolína Burešová, Martin Mareš, Lucie Mohelníková
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 triggery. Rozdíly mezi dialekty SQL.

Dynamický web a PHP *(“Pepíčku, napíšeš mi é-šopík?”)* **PHP**
Jirka Setnička, Karolína Burešová, Jan Kubálek
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 podélně?”)* **KOMP**
Martin Mareš, Michal Vaner
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.

Hardware a operační systémy

Principy počítačů *(“A opravdu uvnitř počítače běhají malí trpaslíci?”)* **HW**
Martin Mareš, Jirka Setnička, Jenda Hadrava
Vydáme se do země skřítků, kteří pohánějí počítače. Počítačové architektury od hodinek 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**
Martin Mareš, Filip Štědronský
Kamarád u černobílého textového okna září blahem. Chcete poznat, proč? Jak UNIX vznikl, k čemu je dobrý a k čemu třeba není. UNIXová filosofie. Kouzlo skriptů. Kouzlo speciálních souborů. Kouzlo propojování programů. Kouzlo nechtěného. UNIX byl napsán v C a C vzniklo pod UNIXem.

Skriptování v shellu *(“man 1 woman … man 2 woman … man group”)* **SHELL**
Filip Štědronský, Jirka Setnička
Spíše motivační přednáška pro ty, kteří shell ještě neviděli, případně jen z dálky. 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 přejmenovávat epizody (legálně ;-)) stažených seriálů dle názvů z Wikipedie, za deset minut zbastlíme zvukem řízenou samospoušť či ušetříme několik set korun za učebnici angličtiny. True story. Některé činnosti vyžadují lidskou nápaditost a vhled. Ty ostatní bychom měli přenechat strojům.

Programování v Linuxu **PLX**
Michal Vaner, Martin Mareš, Filip Štědronský
Jak si program pod Linuxem povídá s operačním systémem, když chce otevřít soubor, přečíst soubor, půjčit trochu paměti a jiná šprťouchlata. 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**
Martin Mareš, Michal Vaner, Jenda Hadrava, Karolína Burešová
Jak funguje Internet a počítačové sítě vůbec: od elektronů v drátech (fotonů v optických kabelech nebo elektromagnetických vln) přes packety 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 – aneb aplikační protokoly TCP/IP *("Pokud jste se zamotali do sítí, tak se vás pokusíme vymotat.")* **NET2**
Martin Mareš, Michal Vaner, Karolína Burešová
Tato přednáška navazuje na „Sítě a Internet" a zaměří se na konkrétní aplikační protokoly nad TCP/IP. Zajímá vás, jak funguje web, pošta, DNS, FTP, nebo třeba Jabber? Poodhalíme roušku tajemství těchto protokolů a když zbude čas, přidáme ještě třeba SIP (protokol pro internetovou telefonii).
Předpoklady: NET

Web uvnitř *("Error 402: Payment Required. Please insert a coin.")* **HTTP**
Martin Mareš, Jirka Setnička, Jan Kubálek, Karolína Burešová
Většina webu je dnes založena na protokolu HTTP, pojďme se podívat, jak funguje uvnitř. Metody GET, POST, ale třeba i PUT. Dohadování o typu dat. Cacheování, revalidace a transformace dat. Křupavé sušenky. Jak se vypořádat s dynamicky generovaným obsahem aneb protokol CGI. Mezi klientem a serverem aneb DNS a virtuální servery. Nakonec do toho všeho přimícháme SSL/TLS a máme HTTPS.

Kryptologie *("Gbgb arav zbp gnwan mcenin.")* **CRYPT**
Martin Mareš, Jirka Setnička, Jenda Hadrava, Karolína Burešová
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.

Síťová (ne)bezpečnost *("Kdo si čte vaše emaily a jak?")* **NEBEZP**
Jirka Setnička, Jenda Hadrava
Stručný úvod do fungování sítě. Praktické ukázky hackování běžné sítě od MITM po převzetí (třeba emailového) účtu. Pomůže nám HTTPS? Jak je to na WiFi? Zastaví nás heslo? Na jak dlouho? A konečně – jak se tomu všemu lze bránit, pokud to vůbec jde.

Teoretická informatika

Složitější složitost * **SLOZ2**
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

Modely počítačů *("Nač Pentium? Máme Turingovy stroje!")* **MODEL**
Martin Mareš, Michal Vaner, Jenda Hadrava
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. Rekursivní funkce, přepisovací pravidla a Minského registrové stroje. Paralelní verze klasických modelů, buněčné a grafové automaty. Trocha esoteriky: reverzibilní algoritmy a dlaždičky v koupelně.

Paralelní výpočty *("Největším nepřítelem lidstva je trojrozměrný prostor.")* **PARAL**
Martin Mareš, Michal Vaner
Když nestihne problém vyřešit jeden procesor, proč jich nepoužít víc? Zkusme na chvíli zavřít oči a představit si, že máme stroj, který umí například pro sečtení N čísel zapnout N procesorů… nebo rovnou N^2, všechny se společnou pamětí a společným programem – teoretikové takovému počítači říkají PRAM. Ukážeme si rychlé paralelní algoritmy všeho druhu: aritmetiku, slévání a třídění, grafové algoritmy, vše v (poly)logaritmickém nebo dokonce konstantním čase. Po probuzení do reality všedního dne trocha praxe: SMP, NUMA, Connection Machine, clustery, koordinované screen savery, FPGA.

Jazyky, gramatiky a automaty * **AUTO**
Martin Mareš, Michal Vaner, Jirka Setnička, Jenda Hadrava, Karolína Burešová
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**

Martin Mareš, Lucie Mohelníková, Jirka Setnička

Teorie grafů trochu teoretičtěji. Různé druhy grafů a jejich vlastnosti. Stromy a lesy. Kreslení grafů jedním tahem. Princip sudosti a skóre grafu. Jak poznat, že dva grafy (ne)jsou isomorfní. Mosty, artikulace a ušaté lemma. Párování, střídavé cesty a Hallova věta.

Rovinné grafy *("Kdo nakreslí pět souvislých států tak, aby každý sousedil s každým, má u mě čokoládu.")* **ROG**

Martin Mareš, Jirka Setnička, Lucie Mohelníková

Povídání o grafech, které jde nakreslit na papír bez křížení hran. Co všechno pro takové grafy platí a jak je poznáme, aniž bychom je museli kreslit. Existuje pouze 5 pravidelných mnohostěnů, a my se o tom pomocí teorie grafů přesvědčíme. Barvení rovinného grafu šesti a možná i méně barvami. Když zbyde čas, zkusíme grafy kreslit i na jiné plochy: kupříkladu Möbiovu pásku, pneumatiku nebo ušatou kouli.

Úvod do teorie čísel *("Po malém fermetu mívám čínský zbytkáč.")* **NUT**

Martin Mareš, Lucie Mohelníková, Karolína 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 specifičtější 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 dál věnovat.

Algoritmy a datové struktury

Datové struktury pro šílence ** *("log log log ... glo glo glo")* **DS3**
Martin Mareš
Ještě důmyslnější datové struktury dle přání posluchačů. Možno servírovat například různé dynamické reprezentace grafů (Sleatorovy-Tarjanovy stromy, ET-stromy, Fredericksonovy topologické stromy), geometrické datové struktury, obecné dynamizační schéma, triky pro malé integery, persistentní datové struktury et cetera.

Nejkratší a jiné cesty * *("Všechny cesty vedou do Horní Dolní, jen některé přes Řím.")* **CESTY**
Martin Mareš, Jirka Setnička, Lucie Mohelníková, Jenda Hadrava, Karolína Burešová
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.

Stringové algoritmy * *("Co se nedá spočítat v lineárním čase, nestojí za to.")* **STRG**
Martin Mareš
Předvedeme všeliké algoritmy na zpracování řetězců, které mají (mimo jiné) společné to, že pracují v lineárním čase: třídění za pomoci kyblíčků, konstrukce suffixových stromů (aneb jak obrátit řetězec naruby) a jejich použití, nebo třeba hledání nejdelšího společného podřetězce dvou řetězců.

Komprese dat *("Jnm idln kpln j nstlčtln.")* **PRESS**
Martin Mareš
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**
Martin Mareš
O algoritmech značně magických a nečekaných. Jak násobit n-ciferná čísla rychleji než v kvadratickém čase. Kouzlo na slévání setříděných posloupností v konstantním prostoru. Isomorfismus stromů pomocí přihrádkového třídění. Bitové kejklířství. Hledání největší díry.

Programovací jazyky

C for wizards * *("`1[x]+++++x[1]`")* **CWIZ**
Martin Mareš
Ponořme se do hlubin Céčka, snad až na samé dno. Typový systém: elementární typy, typové výrazy, automatické konverze a rozpad typů (pole vs. ukazatel). Pořadí vyhodnocování kontra pořadí side-efektů (priority, synchronizační body a volatile). Triky s preprocesorem. Návěští a příkaz switch. Všelijaké zrady (velikosti typů, zarovnání, $(a+b)+c \neq a+(b+c)$, ...). Dialekty Céčka od K&R až po novou normu C11 a různá nestandardní rozšíření jazyka. Proč jsou objekty potřebnější v mysli programátorově než v jazyce a proč je C lepší než C++ ☺
Předpoklady: Povšechná znalost jazyka C.

Černá magie v C++ * *("Je dobré znát, co umí atomová bomba (a její datový typ), abychom ji nechtěli použít.")* **CPP**
Michal Vaner
V C++ jde samozřejmě psát obvyklým způsobem pomocí tříd, polymorfismu a s ruční správou paměti. Ale proč to dělat jednoduše, když to jde složitě? V C++ si můžeme trochu zaprogramovat v době překladu, dělat si seznamy typů, vytvářet lambda třídy, copy-on-write struktury s počítáním referencí... prostě si řekněte, co chcete, napsat to půjde, jen to možná bude práce pro vraha.
Předpoklady: OBJ, TEMPL, staticky alokovaný kyblík

Perl 6 *("Slečno, mohu vám ukázat svou sbírku operátorů?")* **P6**
Martin Mareš
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.

Pokročilé povídání o Pythonu *("`import antigravity`")* **PYTH2**
Filip Štědronský
Povídání o méně zmiňovaných částech Pythonu. New-style classes, dekorátory, metaclasses, generátory, funkcionální styl programování v Pythonu. Jak napsat quicksort jako lambda funkci. Představení zajímavých modulů nejen ze standardní knihovny.
Předpoklady: PYTH

Programování v assembleru **PASM**
Martin Mareš, Jan Kubálek
Jak programovat procesor přímo, aniž by vám do toho mluvily překladače, linkery a podobná verbež. Začneme obecně, ale soustředíme se hlavně na procesory rodiny x86. 32-bitová a 64-bitová instrukční sada, FPU a panoptikum vektorových instrukcí. Rozdíly mezi intelovskou a AT&T syntaxí. Jak spojit assembler s vyššími programovacími jazyky. Optimalizace kódu. Stručný úvod do systémových architektur IA32 a AMD64.

Jazyková Zoo *("Na co GO TO? Máme COME FROM.")* **JZOO**
Martin Mareš
Obecná teorie programovacích jazyků má asi tolik půvabu, jako biologická systematika. Tak se raději pojďme podívat do zoo: poznejme jazyky klasické, experimentální i dočista absurdní. Ada, Céčko a Python (tři pohledy na fungování typů). Praděde-ček všech funkcionálních jazyků LISP (program a data jsou totéž). APL (algebraické inspirace, nebo též průvan ve skladišti písmenek). Forth (zásobníkový předchůdce Postscriptu, ale i javovského virtuálního stroje). Lingua::Romana::Perligata (programovací jazyk, který skloňuje a časuje). Shakespeare, Intercal, Oook! a jiné komedie. Samorozšiřitelné a hybridní jazyky.

Programovací nástroje a techniky

Git a jiné systémy pro správu verzí *("U svatýho tučňáka, kdo sem napsal tohle? Ono to tvrdí, že JÁ?!")* **GIT**
Martin Mareš, Michal Vaner, Karolína Burešová, Jan Kubálek, Jirka Setnička
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 půlením 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**
Michal Vaner, Jirka Setnička
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**
Michal Vaner, Martin Mareš, Jan Kubálek
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**
Martin Mareš, Karolína Burešová
Odložme na chvíli své myše 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**
Michal Vaner, Martin Mareš, Karolína Burešová, Jan Kubálek
Tvrdí se, že číst 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.

High-Performance Computing *("Jak krotit terabyty a jak trilobyty?")* **HPC**
Martin Mareš
Jak vymáčknout z počítače co možná největší výkon. Kdy optimalizovat a kdy raději ne. Jak si program zparalelizovat: aritmetický paralelismus, vektorové instrukce, symetrický i nepříliš symetrický multiprocesing, počítání na clusterech počítačů. K čemu je grafická karta. Lži, zatracené lži a benchmarky a co si z nich vybrat. Jak hledat v terabytovém textu.

Programování na grafické kartě * *("Řídí se to jako raketa – létá rychle, ale nemá volant.")* **GPU**
Michal Vaner
Dnes již není grafická karta jen placka převádějící digitální pixely na analogový signál. Dá se na ní počítat kde co. Zde si představíme trochu OpenCL a zmíníme, že tento ďábelský kus HW umí počítat zatraceně rychle, ale pokud tam uděláme malou chybičku, tak také zatraceně pomalu. Zmíníme, proč tomu tak je, jaké druhy paměti můžeme v programu používat a co je to multiprocesor.

Hardware a operační systémy

Vstupní a výstupní zařízení **IO**
Michal Vaner
Co jde k počítači připojit? To nejsou jen klávesnice a myši. Existují různá další udělátka, jako trackballs, tablety (v původním významu slova), volanty, joysticky, mikrofony a kamery. K výstupním se budou řadit bedničky, monitory, tiskárny a třeba i ty volanty. Mezi méně obvyklé i čtečky čárových kódů, infračervené senzory a dotykové tabule. Jak je to celé připojené a jak to funguje v softwaru. Jak fungovaly myši kdysi a jak dnes. Bude i několik praktických ukázek.

Filesystémy *("Opravdu je FAT tabulka tlustá?")* **FS**
Martin Mareš
Povídání o tom, co leží mezi nulami a jedničkami na plotnách disku a přátelskou adresářovou strukturou našeho OS. Jak funguje FAT a jeho varianty (VFAT, FAT32). Tradiční Linuxové filesystémy od EXT2 k EXT4. Nadějný nový BtrFS, který je možná za pár let nahradí. Co se hodí na SSD.

Digitální elektronika a hradla **DIGI**
Martin Mareš, Jenda Hadrava, Jan Kubálek
Jak fungují digitální elektronické obvody, ze kterých jsou postavené (nejen) počítače. Nuly a jedničky jako napěťové úrovně; kombinační obvody (transistory, hradla, multiplexery), sekvenční obvody (klopné obvody, registry, čítače) a asynchronní obvody. Troška matematiky okolo aneb logické formulky a De Morganovy zákony; proč stačí jenom jeden typ hradel. Třístavová hradla a sběrnice... zde plynule přecházíme v HW.

OpenWRT * *("Linux v každém routeru")* **WRT**
Michal Vaner
OpenWRT je neobvyklá linuxová distribuce určená pro domácí routery. Nemá grafické prostředí a mnoho utilit je takových neúplných, ale zase se to vejde do 4MB flash paměti. Ukážeme si, jak to zkompilovat, nainstalovat a nastavit. A také jak pro to napsat balíček.
Předpoklady: MAKE, UNIX

Správa paměti * *("Když má program sklerózu...")* **MEM**
Michal Vaner
Po chvíli zjistíme, že nám lokální a globální proměnné nestačí a je potřeba paměť alokovat dynamicky. Co všechno si musíme udělat sami a co se děje programátorovi „za zády". Mapování adresního prostoru, ruční alokování a vracení paměti a problémy s tím spojené (chyby programátora), počítání odkazů a daň s nimi spojená (a hele, cyklus), odklízeče odpadu (mark & sweep, kopírovací, generační a jiné triky).

Cache-oblivious algoritmy *("Kešuješ, kešuje, kešujeme")* **CACHE**
Martin Mareš, Michal Vaner
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**
Martin Mareš
Co ten kernel vlastně je, čím se liší programování v kernelu od normálního kódu, jak sobě vlastní kernel postaviti a jak v něm něco opraviti. Kde najít nejnovější zdrojáky a kde najít pomoc, až se něco pokazí.

Sítě a bezpečnost

DNS *("Zeptej se toho serveru vedle")* **DNS**
Michal Vaner, Jirka Setnička
DNS je starý, a přesto moderní protokol. Stojí na něm infrastruktura celého Internetu. Slouží k překladu adres, ale nejen k tomu. Jak zajišťuje spolehlivost, jak bezpečnost. A proč si na něj ani Anonymous netroufnou. V čem je ČR první a co je to digitální lukostřelba.

E-mail *("Drahoušek zákazník.")* **EMAIL**
Michal Vaner
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.

Kryptologie II * *("6140 a184 c9a6 41f1 de99 e733 354a f451")* **CRYPT2**
Martin Mareš
Pokročilejší (dešifruj: zběsilejší) partie vědy kryptologické: utajené výpočty, zero-knowledge proofs, sdílení tajemství, podprahové informace a kvantová kryptografie. Aplikace v reálném životě: digitální peníze, volební systémy. Různé metody útoků na šifry a kryptografické protokoly. Problémy distribuce klíčů a proč se jí raději vyhnout (a jak: Diffie-Hellman key agreement, komutativní šifry). Stručný přehled souvisejících partií matematiky a teorie složitosti.
Předpoklady: Základní povědomí o šifrování (CRYPT) a víra v existenci náhodných čísel

Sledování sítě * *("Nedivte se, že to padá, všichni koukaj na Dalas.")* **SCAN**
Michal Vaner
Jak zjistit, co vám uživatelé dělají na síti. Jak chytit spamera, jak odhalit DDOS, a co jsou jen živelné katastrofy. Jak to dělat doma, v malé síti a jak na 100GBit lince. A co si ještě může administrátor dovolit sledovat v ČR a co v Americe.
Předpoklady: NET, tušení, co je statistika.

Počítačová bezpečnost prakticky * *("K čemu je 4096bitový klíč, když si ho napíšu na papírek pod klávesnici?")* **NSA**
Michal Vaner
Nejslabším článkem bezpečnosti téměř vždy bývá člověk. Na co si dát pozor, když si chráníte data a když posíláte zprávu. Co je to trust model a komu můžete (ne)věřit. Jak udělat bezpečné přihlášení. Co dělat ve chvíli, kdy jsou nějaká data kompromitovaná. A proč zamykat dveře a školit zaměstnance, aby za sebou zavírali a nikoho nepouštěli. Kam data neukládat a jaká data neukládat vůbec.
Předpoklady: Paranoia

Grafika a typografie

Počítačová grafika *("Namaluj mi beránka...")* **GFX**
Martin Mareš, 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 čáry 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.

Barevné systémy *("Co je na konci duhy?")* **COLOR**
Martin Mareš, Jirka Setnička
O podstatě světla a barevného vidění a různých pokusech o reprezentaci barev v počítačích, fotoaparátech, televizích a podobných zařízeních. Systémy RGB, CMY(K), HSV, XYZ, Lab s jejich výhodami i neduhy. „Systém" Pantone. Reálné kontra imaginární barvy aneb proč nejde vyfotit duha.

PostScript a PDF *("Vy obrázky malujete? To my je programujeme...")* **PS**
Martin Mareš
Jemný úvod do jazyka určeného k tisku grafiky a textu. Základní principy, řídící konstrukce a datové struktury, cesty a kreslení objektů, transformace souřadnic, DSC komentáře. PDF (Portable Document Format) coby mladší a deklarativnější bráška PostScriptu. Různé druhy fontů (např. Type1, TrueType) a jak fungují.

MetaFont, MetaPost *("Teď ten obrázek takhle zkroutím a pak ho přeložím.")* **MF**
Lucie Mohelníková, 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**
Martin Mareš, Karolína 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 TEXnician.")* **TEX**
Martin Mareš, Lucie Mohelníková, Jirka Setnička, Karolína Burešová
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 TEXu od hladké sazby knihy až po zběsilosti hraničící s programováním. Pozornost věnujeme i zdrojům informací a rozdílům mezi různými dialekty TEXu.

OpenGL * *("Je libo krychličku nebo díru do ní?")* **GL**
Michal Vaner
Základy OpenGL, jak vytvořit 3D svět. Kreslení trojúhelníků a jiných úhelníků. Barvy, textury a světlo. Průhlednost a triky s ní. A triky bez ní. Jak udělat žulový náhrobek s vyrytým reliéfním nápisem na 6 obdélníků a hladkou kouli na 16. Jak udělat oheň či vodotrysk jako živý. Náznaky, co vše může dnešní grafická karta dělat a jak moc jsou výrobci her lemry líné.
Předpoklady: GFX

Teoretická informatika

Třídy složitosti * **SLOZ3**
Martin Mareš, Michal Vaner
Složitost opravdu důkladně: nejrůznější třídy složitosti a vztahy mezi nimi. Vztahy mezi časem a prostorem, odstraňování nedeterminismu a Savitchova věta. Jak víme, že všechny třídy nejsou stejné: dolní odhady a věty o hierarchii. Stroje s kvantifikátory, třída PSPACE a polynomiální hierarchie. Pravděpodobnostní třídy složitosti. Orákula a neuniformní složitost.
Předpoklady: SLOZ2

Pravděpodobnost a algoritmy *("Nejen že Bůh hraje v kostky, ale ještě při tom občas švindluje!")* **PPALG**
Martin Mareš
K čemu jsou při programování dobrá náhodná čísla a jak je generovat. Algoritmy pravděpodobnostní a randomizované, průměrná časová složitost a její koncentrace. Míchání karet. Proč používat a proč nepoužívat Quicksort. Inkrementální algoritmy (třeba na konvexní obal), vyhledávání v poli v konstantním čase za pomoci hešování, konstrukce perfektního hešování, randomizované datové struktury (skip listy a treapy). Interaktivní protokoly aneb jak vyhrát nad falešným hráčem.

Vyčíslitelnost ** *("S Halting problémem na věčné časy!")* **VYCIS**
Martin Mareš
Některé problémy se dají vyřešit snadno, jiné obtížněji a některé dokonce vůbec. Obecněji: Ať si vymyslíte jakýkoliv rozumný programovací jazyk, vždycky existuje problém, který se v něm nedá vyřešit. Jak se ale dokazuje, že něco nejde? Matematický pohled na výpočetní modely a univerzální stroje, rekurzivně spočetné a rekurzivní množiny a funkce. Halting problem a diagonální důkazy.

Omezené třídy grafů ** *("Nejdelší cesta ve stromu? A co je za problém?")* **OTG**
Michal Vaner
Některé problémy nad grafy jsou těžké, ale pokud si omezíme grafy, které můžeme dostat, hned je to jednodušší. Co je to graf omezené stromové šířky, a jak na něm najít hamiltonovskou kružnici v polynomiálním čase. Silně teoretická přednáška.
Předpoklady: GA, SLOZ2

Programování s omezujícími podmínkami *("Celé prázdniny budu plánovat a řešit sudoku.")* **CSTR**
Michal Vaner, Lucie Mohelníková
Trochu jiný přístup k obtížným úlohám. Některé úlohy sice vypadají, jako by se za dobu existence vesmíru nedaly vyřešit, nicméně pro rozumně velké vstupy to přesto potřebujeme. Jak backtrackovat rychleji a radostněji – backjumping, backmarking, limited discrepancy search, a jak neprobírat úplné nesmysly – hranová konzistence, konzistence po cestě, bodová konzistence.

Aplikace informatiky

Počítačová lingvistika *("Jsou bramborové knedlíky plněné bramborami?")* **CMPLING**
Karolína 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**
Karolína 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 nevadí, že Vim by v některých testech rozhodně neuspěl.

Čárové kódy *("Jak naučit počítače číst láhve od Coly")* **BAR**
Martin Mareš
Čárové kódy dnes potkáváme na každém kroku, ale jak doopravdy fungují? Prozkoumáme klasické jednorozměrné kódy (UPC, EAN, Code39, Code128), jakož i novější dvojrozměrné (QR, Aztec, DataMatrix). Kódovací a dekódovací algoritmy plus trocha matematiky okolo zabezpečení proti chybám. Další počítačem čitelné značky: RFID, bíle křížky na asfaltu, ...

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.

Real-time osobní komunikace po Internetu *("Zítra v 5 v matfyz@conf.netlab.cz")* **VOIPIM**
Michal Vaner
Probereme sbírku různých komunikačních systémů, které usnadňují život na internetu. Jak si posílat zprávičky, počínaje starým dobrým talkem, přes IRC až po XMPP (to, na čem běží všechny jabbery, google talky i facebook chaty). Přejdeme i k telefonování, zmíníme SIP, Skype a Jingle. Jak to funguje uvnitř? Čemu se vyhnout, pokud chceme komunikovat i za 20 let, co se dá odposlouchávat a k čemu se hodí otevřenost? A proč to tak moc vadí telekomunikačním společnostem?
Předpoklady: Základní povědomí o fungování počítačových sítí

Matematické přednášky

Logika *("Tato věta sem nepatří.")* **LOGI**
Martin Mareš
Pokud budeme v životě věřit všemu, co je „přeci zřejmé", dostaneme se brzy do potíží a v matematice to platí dvojnásob. Ale co s tím? Přírodní vědy si vymyslely verifikovatelné experimenty a matematici logiku a dokazování. Co je to výrok, co jeho důkaz a proč se axiomy nedokazují. Jenže jak si je zvolit? A jak se z toho všeho postaví celá matematika? A bude vůbec matematika někdy celá? Studená sprcha pana Gödela coby sebevražedné dovršení snahy získat dokonalý jazyk. Logika coby hra a problém líného profesora. Důkazy boží existence a neexistence.
Předpoklady: LOGI

Pravděpodobnost *("Většina lidí má nadprůměrný počet rukou.")* **PP**
Martin Mareš
Toto je přednáška o základech teorie pravděpodobnosti a statistiky. Dozvíte se, co to je podmíněná pravděpodobnost, rozdělení, střední hodnota nebo rozptyl, jak se to všechno počítá a k čemu je to dobré. Součástí přednášky bude i několik zajímavých příkladů z praxe a krátký kurs přežití ve světě plném chybných statistik.

Úvod do Ramseyovy teorie * *("Dejte mi dostatečně velký objekt a já v něm najdu nějaký řád.")* **RAMS**
Martin Mareš, Jirka Setnička
Hříčka: ve společnosti šesti lidí vždy existují tři, kteří se navzájem znají, nebo neznají (ověřte ručně). Obecněji, pro libovolné „tři" existuje „šest" tak, že shora uvedené tvrzení platí. To je jedna z Ramseyových vět, které říkají, že v každém dostatečně velkém objektu vždy existuje nějaký stejnorodý podobjekt. Jednoduchá tvrzení Ramseyova typu, Ramseyova věta pro grafy dvou a více barev, pro systémy p-tic, nekonečná verze a aplikace. Populárně řečeno, chaos to má těžké.

Teorie množin a matematika nekonečen * *("Je Vlk nedosažitelný kardinál?")* **TEMNO**
Martin Mareš
Teorie množin tvoří páteř veškeré matematiky. Pomocí množin se totiž modelují veškeré objekty, které se v matematice vyskytují. Celou teorii prostupuje magický pojem *nekonečno.* Jakým způsobem se tohoto, pro spekulativní mysl ošidného, termínu zhostila moderní matematika? Množiny a jejich velikosti. Cantorův diagonální trik. Ordinály a houšť kardinálů. Potenciální kontra aktuální nekonečno. Myslíte si, že máte dobrou představu o tom, co jsou přirozená čísla? Možná vás z ní vyvedeme. A co teprve reálná čísla. Problematika volby axiomů determinovanosti versus výběru.

Barevnost grafů * *("Bílá, modrá, červená, co to pro graf znamená?")* **BAGR**
Michal Vaner, Lucie Mohelníková
V teorii grafů zaujímá významné místo problém barevnosti grafu, tedy přiřazení co nejmenší počtu barev vrcholům tak, aby se hranami dotýkaly pouze různobarevné vrcholy. Aplikace problému v informatice je nasnadě. Ukážeme si několik zajímavých teoretických výsledků. Obarvení některých druhů grafů, $L_{2,1}$ barevnost aneb problém vysílačů, vybíravost, kruhová barevnost a další.

Lineární algebra **LA**
Martin Mareš, Lucie Mohelníková, Jan Kubálek, Karolína Burešová, Jenda Hadrava
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.

Lineární algebra pro pokročilé **LAFA**
Jan Kubálek
Volně navážeme na LA. Ukážeme si, co je to spektrum nebo nilpotentní matice. Proč lze ke každé čtvercové matici najít podobnou matici, která bude horní trojúhelníková nebo proč každý matfyzák snídá koláče upečené v bilineární formě. Nebo snad chcete vědět, jak Rusko málem zničilo planetu setrvačníkem?
Předpoklady: Doporučené jsou základní znalosti z LA.

Lineární programování ** **OPT**
Lucie Mohelníková
Řešení problémů pomocí soustavy lineárních nerovnic (lineárním programováním). Naučíme se tak řešit některé grafové, ale i jiné, problémy. Řekneme, jaké metody se k tomu používají a jak jsou efektivní oproti běžnému přístupu.
Předpoklady: LA

Teorie (vesměs samoopravných) kódů *("f y cn rd ths, y wll b gd cmptr prgrmmr!")* **KODY**
Martin Mareš
Jak komunikovat po lince, která průměrně každý k-tý bit přenese špatně? K tomu se hodí teorie samoopravných kódů, která nás naučí: vzdálenost slov a jejich souvislost s detekcí a opravou chyb, paritní a lineární kódy, perfektní kódy, Reed-Solomonovy a vůbec polynomiální kódy a několik dolních odhadů nádavkem. A jak s teorií kódů souvisí třeba čeština?

Komplexní a komplexnější čísla *("$1 = \sqrt{1} = \sqrt{(-1)(-1)} = \sqrt{-1}\sqrt{-1} = i \cdot i = i^2 = -1$. Huh?")* **CPLX**
Martin Mareš, Jan Kubálek, Lucie Mohelníková, Jenda Hadrava
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.

Teorie čísel a RSA * *("$2^{67} - 1 = 193\,707\,721 \cdot 761\,838\,257\,287$")* **NUT2**
Martin Mareš, Karolína Burešová
Pokračování teorie čísel, které nás dovede až k RSA – asi nejpoužívanějšímu asymetrickému šifrovacímu algoritmu dnešní doby. Počítání modulo složené číslo a Eulerova věta. Jak RSA funguje, proč funguje a jestli bude ještě fungovat. Generování klíčů, faktorizace kontra testování prvočíselnosti. Časová složitost aritmetiky.

Prvočíselné věty * **NUT3**
Martin Mareš
Věty o rozložení prvočísel jsou tradičně považovány za jednu z nejmysterióznějších oblastí teorie čísel. Zde ukážeme, jak některé z nich odvodit snadným pozorováním vlastností kombinačních čísel: rozbor časové složitosti Eratosthenova síta, Bertrandův postulát („Mezi n a $2n$ je aspoň jedno prvočíslo."), hustota prvočísel.

Fourierova transformace * **FFT**
Martin Mareš, Jenda Hadrava
Chytrý trik pana Fouriera patří již dávno k matematické a fyzikální klasice. Převapivě se ale hodí i při programování: rychlé násobení polynomů a dlouhých čísel (dokonce v lineárním čase), digitální zpracování zvuku a obrazu (spektrální analýza či třeba komprese).
Předpoklady: Základy komplexních čísel (CPLX)

Teorie nemožného * *("Neexistence důkazu není důkazem neexistence. Dokažte.")* **NONEX**
Martin Mareš
Existenci slona v Africe snadno dokážete tím, že ho přivedete. Jak ale ukázat, že tam žádný slon není, případně že sice je, jenže ho nejde najít pomocí pravítka, kružítka a jeepu? Přímo se to dělá těžko, ale existuje spousta krásných triků, jak neřešitelnost problémů dokazovat. Nesložitelné hlavolamy, nerozvázatelné uzly, nepopsatelná čísla, neroztřetitelné úhly, nealgoritmické problémy a jiné slasti nekonstruktivní matematiky. Jak naopak ukázat, že něco existuje, aniž bychom věděli, jak to vypadá?

Úvod do klasické analýzy * *("Dej pokoj a už mě konečně zinfinitizuj")* **KA**
Jan Kubálek
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**
Jirka Setnička
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.
Předpoklady: Prostorová představivost výhodou, pravítko a kružítko rovněž, koulítko a rovinítko netřeba.

Kombinatorické struktury * *("BIBy, BIBDy a jiná chlupatá zvířátka")* **KST**
Lucie Mohelníková
Kombinatorika v sobě skrývá různa (ne)tradiční schémata, o jejichž použitelnosti bychom se mohli přít hodiny a hodiny. Ukážeme si, co jsou to latinské čtverce, projektivní roviny a jak souvisí s BIBDy (bloková schémata). Dokážeme si větu, kterou budeme mít problém si vůbec zapamatovat, a povíme si něco o Steinerových systémech trojic. Přednáška bude hodně matematická a absolutně nepředstavitelná.

Catalanova a Fibonacciho čísla * *("1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, ?")* **CAT**
Lucie Moheníková, Martin Mareš
Kolik existuje binárních stromů? Kolika způsoby jde uzávorkovat výraz? A koika způsoby projít čtvercovou mřížku, aniž bychom překročili úhlopříčku? Kam oko pohlédne, všude se skrývají Catalanova čísla. Kromě případů, kdy za ně zaskakují čísla Fibonacciho. Povídání o dvou zajímavých posloupnostech a jejich početném příbuzenstvu. Dlouhá cesta od hezkého vzorečku k rychlému algoritmu.

Fyzikální přednášky

Podzimní obloha **SKY**

Martin Mareš

Pozorování podzimní hvězdné oblohy spojené s astronomickým minikursem. Od antických a ještě starších bájí k modernímu příběhu o Velkém Třesku a naopak od celkem seriózní vědy k rozmarnému filosofování o světě a našem místě v něm. Hvězdáři a hvězdopravci, „Už staří Řekové…“, měření a vážení na dálku, vývoj hvězd a kosmologie, antropický princip, kdo schvaluje fyzikální zákony? Jak se podle hvězd orientovat a jak fungují sluneční a třeba i měsíční hodiny.

Předpoklady: Počasí dovolí. Měsíc nejlépe v novu.

Keplerovy zákony trochu jinak * *(“aneb jak dostat kosmonauta na měsíc”)* **KEPP**

Jan Kubálek

Jsou Keplerovy zákony přesné, nebo se jedná o aproximaci? Ukážeme si, jak odvodit Keplerovy zákony z čistě matematického základu, jak vypočítat dráhu rakety letící k měsíci nebo proč planety obíhají po elipsách.

Předpoklady: Znát pojem derivace (KA) a vektoru (LA).

Elektřina a magnetismus * **ELMAG**

Jan Kubálek

Tak jak se učí na matfyzu. Co to jsou Maxwellovy rovnice, jak vznikly a co znamenají všechna ta písmenka kolem.

Předpoklady: Znát pojem derivace (KA) a vektoru (LA).

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.

Lingvištika *("Přísudek je v této větě podmět.")* **LING**
Martin Mareš
Převážně nevážné 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.

Fonetika *("Pojďte, zachrochtáme si spolu!")* **FON**
Martin Mareš
Malá inventura zvuků, které lidé dovedou vytvářet, a jejich použití v komunikaci. Různé způsoby vytváření a modulace zvuku. Kolik různých B dokážete říci? Fonetické kontrasty a co si z nich různé jazyky vybraly. Rázy, polosamohlásky a jiní obyvatelé polosvěta. Přízvuk kontra délka. Asimilace, přehlasování a další „principy líné huby." Vše prakticky procvičíme.

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 sepsaný Úvod do matfyzáka a zazpíváme pár matfyzáckých písní. Zbytek už bude záležet na tom, co budete chtít slyšet.

Sbírka brouků *("No keyboard found. Press any key to continue.")* **BUG**
Michal Vaner
Neuspořádaná přehlídka, jaké chyby kdo kdy udělal, kdy se z nich stal stal standard a jak se to komu povedlo využít. Jak některá taková chyba stála miliony dolarů, či dlouhé hodiny hledání. Očekává se přispění posluchačů.

Nečisté obchodní praktiky *("Pokud odejdeš, pošleme ti z tvých dat napřed malíček, potom prsteníček. . . ")* **SCUM**
Michal Vaner
Některé programy a služby na internetu jsou udělané tak, aby z nich měl uživatel užitek. Některé také tak, aby z nich měl užitek autor. A některé tak, aby z nich měl užitek jen autor a uživatele pokud možno odrbal. Mezi ně zajisté patří různé viry a trojské koně, ale také mnoho legálních věcí. Jak zavírá uživatele do ohrádky takový Microsoft či Apple, jak takový Google a Facebook?

Orientace **ORI**
Martin Mareš, 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 časomíry 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.

OpenStreetMap *("Mapa, ve které je i má oblíbená lavička")* **OSM**
Michal Vaner, Martin Mareš
Dříve se ke kreslení map používaly pastelky, dnes k tomu lze použít také počítač. A v době internetu to i sdílet. Co se stane, když pustíme desetitisíce lidí, kteří začnou kreslit zároveň? Co použít k zaznamenání vlastního domu, kde sebrat řeky a jak z toho nakonec udělat turistickou mapu či autonavigaci? Aneb hračka technologického nadšence do terénu.

Čaj *("Jak vypadá odvar z nezralých pražců?")* **TEA**
Martin Mareš
Pojďme usednout k šálku lahodného čaje a povídat si o tom, co se v něm skrývá. Kde se čaj vzal, kde se pěstuje, jak se zpracovává a jak ho připravovat. Trocha čajového zeměpisu, dějepisu i čajové chemie a čajové kultury. Též o všelijakých substancích čaji podobných.

Programování v týmu *("Přiznejte se, kdo z vás ten bug neopravil!?")* **TEAM**
Michal Vaner, Martin Mareš
Když je program na jednoho člověka moc velký, začne na něm pracovat lidí více. Tato přednáška bude pojednávat o tom, jak takový tým lidí uřídit, od dobrovolnických neřízených projektů, po hierarchie. Které nástroje se k tomu hodí a co dělat, když je tým rozložen do několika časových pásem. A co plyne z toho, že devět žen neporodí dítě za jeden měsíc?

Prezentační dovednosti *("Takže, milánkové, teď vám něco řeknu, takže budete hezky poslouchat.")* **SPEAK**
Michal Vaner
Pokud chcete přednášet na soustředění KSP, prezentovat plán představenstvu společnosti či oslovit voliče, potřebujete se postavit před dav a něco mu sdělit. Ukážeme si základy, jak na to a jak to nepohřbít úplně. Řekneme si, k čemu je potřeba obsah, jak zaujmout a jak pracovat s pozorností publika. Kdy cheme použít prezentaci a kdy to raději nedělat.
Předpoklady: Drzost

Email Best Practices *("Proč si výhru nevyzvedl, když na něj blikala, svítila a pípala?")* **EBP**
Michal Vaner
Jak správně komunikovat emailem? Stejně jako papírové dopisy, i ty elektronické mají nějakou kulturu a měly by dodržovat základní štábní kulturu. Jaká pravidla dodržovat a co naopak nikdy nedělat, aby se emaily příjemci dobře četly? Jak se vyznat v stovkách emailů denně a nezbláznit se z toho? Jak psát známým, jak obchodním partnerům, a jak na konferenci? A co dělat, aby příjemce email okamžitě zahodil? Na tyto otázky si odpovíme a předvedeme i několik ukázek.

Základy první pomoci *("Jak někomu zachránit život a jak málo k tomu stačí")* **ZDRAV**
Jirka Setnička, Karolína 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**
Karolína 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.

Evoluce *("Ale vždyť je to jen teorie!")* **EVOL**
Filip Štědronský
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 Štědronský
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.

Autonomní roboti *("Proč se točí na místě, když má jet rovně!?")* **ROBOT**
Jenda Hadrava
Jak postavit robota a jak jej naprogramovat? Povídání na pomezí virtuálního a reálného světa. Čím robot vidí okolí, jak určuje svoji pozici a jak se pohybuje? Povíme si také, kterak chytrý software může nahradit špatný hardware (a naopak).

Vytvořující funkce **VYTF**
Jenda Hadrava
Vytvořující funkce jsou silný nástroj pro zjišťování počtu možností, obvykle daných nějakou rekurencí, a tvorbu vzorečků. Základní myšlenkou je představit si polynom, jehož koeficienty jsou hledané výsledky, a z něj pak odvodit vytvořující funkci ve tvaru, ze kterého dokážeme zkonstruovat vzoreček. Tímto způsobem můžeme například odvodit vzoreček pro Fibonacciho či Catalanova čísla a mnohé další.

Filosofie programovacích jazyků *("Your programming language sucks!")* **YPLS**
Martin Mareš & Filip Štědronský
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.

Abecední seznam přednášek

Základní přednášky

Rozšiřující přednášky

Půlnoční přednášky