

PODZIMNÍ SOUSTŘEDĚNÍ KSP 2011 – 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 to k přednáškám připište, 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.

Informatické přednášky – teoretické

Jak vypadá řešení (“*Jen jeden bod, když jsem napsal 18 stránek?*”) **SOL**

Jak má správně vypadat řešení KSP? Na co si dát pozor, co je úplně špatně a za co organizátoři strhávají body a sobě vlasy. Přednáška, která by mohla pomoci i mnohým déle aktivním řešitelům.

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 třídících) algoritmů a výpočet jejich složitosti.

Složitější složitost * **SLOZ2**

Trochu hlouběji o složitosti: amortizovaná časová složitost, dolní odhady, nedeterministické výpočty a třída NP, NP-úplné problémy a příklady redukci.

Předpoklady: SLOZ

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

Základní algoritmy **ZALG**

Základní výbavou každého informatika jsou různé standardní algoritmy, zde si ukážeme ty nejdůležitější z nich: Třídící algoritmy včetně vnějšího třídění. Trocha rekurze: hledání mediánu nebo obecněji k -tého nejmenšího prvku v lineárním čase. Aritmetika s dlouhými čísly. Železničářský algoritmus na vyhodnocování výrazů.

Ne až tak základní algoritmy * (“*Pokročilá technologie není rozlišitelná od magie.*”) **NZALG**

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 kejklřství.

Výčísitelnost ** (“*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.

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. Union-Find problem, hledání minimální kostry.

Prohledávání do hloubky **DFS**

Nášup grafových algoritmů, zejména různé chytré aplikace prohledávání do hloubky: topologické třídění, hledání mostů a artikulací, komponenty silné souvislosti, kreslení grafů jedním tahem.

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

Martin Mareš

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^*). Souvislosti s násobením matic: transitivní uzávěr, Seidelův algoritmus, Kleeneho algoritmus a regulární výrazy.

David Marek

Jak se píše umělá inteligence do strategií? A co má společného s navigacemi v autech? Zjistíme, že existuje mnohem více prohledávacích algoritmů než jen DFS a BFS. Projdeme prohledávací algoritmy od naivních přes informované (povíme si o heuristikách) až po online (zabrouzdáme i do genetických algoritmů). Ve zkratkách si povíme o BFS, DFS, UCS, DLS, ID, BS, A*, IDA*, MA*, HC, LBS, GA, ODFS, LRTA*.

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

TOKY

Martin Mareš, Michal Vaner, Martin Böhm, Pavel Veselý

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)

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

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 Smrt. Klasické struktury jako pole, seznamy, vyhledávací stromy (vyvážené, AVL, a - b , splay), haldy (binární a obecně regulární) a v neposlední řadě hashování.

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

DS2

Michal Vaner, Martin Böhm, Martin Mareš

Důmyslnější datové struktury: trie, splay stromy, BB- α stromy; geometrické struktury pro lokalizaci bodů v rovině; binomiální a Fibonacciho haldy, leftist haldy a 2-3 haldy. Též několik přátelských randomizovaných datových struktur: skip listy a treapy.

Datové struktury pro šílence **

DS3

Martin Mareš, Martin Böhm

Ještě důmyslnější datové struktury dle přání posluchačů. Možno servírovat například: dynamické reprezentace grafů (Sleator-Tarjanovy stromy, ET-stromy, Fredericksonovy topologické stromy), vícerozměrné datové struktury (zobecnění vyhledávacích stromů a intervalových stromů), obecné dynamizační schéma, triky pro malé integery, persistentní datové struktury et cetera.

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

ITREE

Karel Tesař

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

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, Jan Matějka

Vyhledávání čehokoliv ve velkém množství textu. Prostá vylepšení hledání hrubou silou – Karp-Rabin, Boyer-Moore. A algoritmy chytřejší – Morris-Pratt, Knuth-Morris-Pratt, Aho-McCorasicková. Konečné automaty teoreticky i prakticky, regulární a „regulární“ výrazy.

Stringové algoritmy * (“Co se nedá spočítat v lineárním čase, nestojí za to.”)

STRG

Martin Mareš, Martin Böhm

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.

Předpoklady: REGEX aspoň zhruba

Kompres 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).

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, časová složitost v průměrném případě. Proč používat a proč nepoužívat Quicksort. Inkrementální algoritmy (třeba na konvexní obal), vyhledávání v poli v konstantním čase za pomoci hashování, konstrukce perfektního hashování, randomizované datové struktury (skip listy a treapy). Interaktivní protokoly aneb jak vyhrát nad falešným hráčem. Problém studny na Pražském hradě. Míchání karet.

Paralelní výpočty (“*Největším nepřitelem 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.

Kryptologie (“*Gbgb arav zbp gnwan mcenin.*”)

CRYPT

Martin Mareš, Karel Tesař, Pavel Čížek

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 jak si hodit korunou po telefonu. Předvedeme nerozluštitelnou šifru a dokonce to o ní i dokážeme.

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

Programování s omezujícími podmínkami (“*Celé prázdniny budu plánovat a řešit sudoku.*”)

CSTR

Michal Vaner

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 úplně nesmysly – hranová konzistence, konzistence po cestě, bodová konzistence.

Informatické přednášky – programovací jazyky a techniky

Programování v jazyce C

C

Michal Vaner, Jan Matějka, Pavel Veselý

Datové typy jazyka C, programové konstrukce, základy práce s ukazateli. Seznámení se standardními knihovnami jazyka C.

C for wizards * (“*1[x]+++++x[1]*”)

CWIZ

Martin Mareš

Céčkové speciality aneb všechno, co jste chtěli o Céčku vědět, ale nebylo se koho zeptat. Pořadí vyhodnocování, side effecty, sequencing pointy, funkce s proměnným počtem parametrů, preprocesorové triky, celá pravda o vztahu pointerů a polí, o jménech typů a o příkazu switch; alignment, NULL, void, volatile. Všelijaké zrady (velikosti typů, $(a + b) + c \neq a + (b + c)$, znaménka ...). Dialekty Céčka od K&R až po (staro)nový standard C99 a různá nestandardní rozšíření jazyka. Proč jsou objekty potřebnější v myslí programátorově než v jazyce a proč je C lepší než C++ ☺

Předpoklady: Povšechná znalost jazyka C.

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

OBJ

Michal Vaner, Pavel Veselý

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.

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 řeknete, co chcete, napsat to půjde, jen to možná bude práce pro vraha.

Předpoklady: OBJ, TEMPL, staticky alokovaný kyblík

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

CIS

Pavel Veselý

C# je moderní objektově orientovaný jazyk, který za deset let svého bouřlivého vývoje dostal do vínku některé funkcionální rysy. Mimo popisu základních konstrukcí, tříd a jejich součástí si projdeme zajímavé části dotnetí Base Class Library.

.NET Framework (“*from p in prednasky where p.Name == ".NET Framework" select p.Content;*”)

DNET

Pavel Veselý

Co je to .NET Framework, jak funguje a k čemu to celé je a není vhodné. Co zajímavého lze najít v knihovnách. Novinky v .NETu v posledních letech: jak vytvářet aplikaci zároveň pro PC i web, proč se vyplatí znát SQL při parsování dokumentů XML, funkcionální jazyk F#, novinky v C#, např. dynamický typ, nepovinné parametry ... (zbytek seznamu vymazán šotkem, prý byl moc dlouhý).

- Java** **JAVA**
David Marek
 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 preprocessor. 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. Co stojí proces, co vlákno. Jaké problémy nastávají ve chvíli, kdy spolu dvě vlákna mají komunikovat. Problémy s nezamknutou pamětí, co je mutex, semafor, podmínková proměnná, deadlock a co se nad tím dá postavit. Jak některé jazyky s tímto pomáhají a jak ne. Shrnutí, k čemu se taková vlákna v praxi hodí a kdy je lepší se obejít bez nich.
Předpoklady: Trochu představy o hardwaru
- Perl** (*“Jak Pejsek a Kočička vymýšleli programovací jazyk”*) **PERL**
Martin Mareš, Michal Vaner, Jan Matějka
 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 jedinečně milovat nebo nenávidět, nic mezi tím. Malé ochutnání Perlu6, jazyka (snad už nepřilíš vzdálené) budoucnosti.
- Python** (*“print "Ffff".decode("rot13")”*) **PYTH**
Michal Vaner, David Marek
 Základy programování v Hroznýši (Pythonu), syntaxe, datové (ne)typy, funkce, třídy, moduly aneb všechno je slovník nebo prvek slovníku (nebo oboje). Výhody interaktivního interpretu.
- Pokročilé povídání o Pythonu** (*“import antigravity ”*) **PYTH2**
David Marek
 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
- Logické programování** (*“Detektivem za 90 minut.”*) **LOGP**
Jan Matějka, David Marek, Pavel Veselý, Pavel Čížek
 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.
- Funkcionální programování** (*“Naše prášky nemají vedlejší účinky.”*) **FUNC**
Martin Böhm, David Marek
 Nenáročná povídání o tom, co je na funkcionálním programování nové, co skvělé a co méně skvělé. Side-effects a proč je dobré být líný. Práce s funkcemi, specializace funkcí, currying. Jak se v tom všem neztratit aneb mapy. Monády. Jako další chod doporučujeme HASK.
- LISP** (*“Lots of Irritating Superfluous Parentheses?”*) **LISP**
Martin Mareš
 Lehký úvod do funkcionálního programování a jazyků z lisovské rodiny (Common Lisp, E-Lisp, Scheme, KSP Lisp atd.). Všechno je funkce, zbytek jsou seznamy (a konec konců funkce je také druh seznamu). Proměnné aneb příběh se nemění, jen příjmení a jména. Jak se programuje v Lispu a jak se programuje Lisp.
Předpoklady: Netrpět uncinofobií (((to jest chorobným strachem ze závorek)))
- Haskell** (*“Pro ty, kdo uncinofobií trpí”*) **HASK**
Michal Vaner, Jan Matějka, Martin Böhm
 Základní kurz Haskellu – moderního funkcionálního jazyka. Na skladě máme skoro všechno, co měl Lisp, o zbytku ukážeme, že mít to by byla chyba; a samozřejmě spoustu věcí navíc. Základní konstrukce, typový systém, třídy a jak se obejít bez výjimek a speciálních případů, vstup a výstup. Pokud zbude čas, tak také trochu bezpečného vícevláknového a paralelního programování.
- Jazyk SQL** (*“SELECT something FROM knowledge LIMIT 45min”*) **SQL**
Jan Matějka
 Jazyk SQL a jeho aplikace. Jak ušetřit skriptu práci a sobě čas, aneb jak se zeptat rovnou na to, co chci vědět. K čemu se hodí složený dotaz a klíčové slovo JOIN. Kam až si můžu dovolit zajít, když nevím, na kterém systému to poběží.

Jazyk XML a související technologie (“<xml style="vesele"/>”)

XML

David Marek

Povíme si, co je to jazyk XML, jak vznikl a k čemu je dobrý. Zároveň probereme nástroje pro práci s XML dokumenty, jejich načítání (parsování), generování, validaci atd. Z pokročilejších technik pak ukážeme vyhledávání v XML pomocí jazyka XPath a transformace XML dokumentů pomocí XSLT. Zbyde-li čas, ukážeme si i několik konkrétních aplikací (XHTML, WML, SVG, ODF ...).

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

GIT

Martin Mareš, Michal Vaner, Jan Matějka

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 pŕlením 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.

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

STYLE

Michal Vaner, Jan Matějka, Martin Mareš

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. Určeno především začátečníkům a zapřísáhlým teoretikům.

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 vrácení 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).

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

MAKE

Michal Vaner, Jan Matějka

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.

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

GDB

Michal Vaner, Jan Matějka, Martin Mareš

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, Jan Matějka

Odložme na chvíli své myši a pojdme 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.

Portabilní programování (“Šel si program na vandr. . .”)

PORT

Martin Mareš

Většina programátorů dřív nebo později zjistí, že počítačový svět nekončí hranicí jejich monitoru a že je mnohem rozmanitější než nekonečné zelené pláne windowsové pracovní plochy. Jenže jak se v takovém světě domluvit a jak psát programy, aby fungovaly všude? Na co se dá spolehnout a na co ne, jaké se hodí znát jazyky a jaké knihovny k nim. K čemu jsou dobré standardy a k čemu `configure`. Proč je někdy potřeba vynalézat kolo. Za rok se vrátím aneb jak (nechat) program udržovat.

Programování v Linuxu

PLX

Jan Matějka, Michal Vaner, Martin Mareš

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 odchyťovat 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.

Programování ve Windows

WIN

David Marek

I když to možná bude znít neuvěřitelně, tak jádro Windows je napsáno vcelku pěkně. Podíváme se jak funguje kernel a jak se dají psát systémové programy v C. Windows API nabízí spoustu vymožeností, které jsou na dosah ruky (např. pracovní fronty a jednoduché asynchronní I/O, díky kterým je jednoduché napsat zpracování mnoha událostí na více jádrech). Také zabrouzdáme do psaní kernel modulů a ukážeme si, jak se píší file system filtry, které jsou využívány v antivirech.

High-Performance Computing (“Jak krotit terabyty a jak trilobyty?”)

HPC

Martin Mareš

Jak vymáčknot z počítače co možná největší výkon. Kdy optimalizovat a kdy raději ne. Jak si program zparallelizovat: aritmetický paralelismus, vektorové instrukce, symetrický i nepřilíš symetrický multiprocessing, 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.

Dynamický web a PHP (*“Pepíčku, napíšeš mi é-šopík?”*)

PHP

Jan Matějka, Pavel Taufer

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

Programování v assembleru

PASM

Martin Mareš

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.

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 multiprocessor.

Evoluční algoritmy (*“thisAlgorithmBecomingSkynetCost = 999999999”*)

EVA

David Marek

Jaký je správný postup pro vytvoření 7-roach rushe v Starcraftu, se dozvíme právě na této přednášce. Samozřejmě začneme pěkně od začátku, řekneme si, co je genetický algoritmus, jak jej napíšeme, k čemu se hodí. Uvidíme, že je výhodné se inspirovat v přírodě, občas si ji můžeme i trochu upravit (povolit žirafám prodlužovat si krky během života). Dostaneme se i k trochu vzdálenějším tématům, jako jsou flocking behaviours anebo optimalizace s pomocí mravenců.

Informatické přednášky – hardware, operační systémy a spol.

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

HW

Martin Mareš

Vydáme se do země skřítků, kteří pohánějí počítače. Počítačové architektury, jejich historie (plná omylů) 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.

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

MODEL

Martin Mareš, Michal Vaner

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, nedeterministický, univerzální a paralelní, orákula, Random Access Machine (RAM), Parallel RAM, Pointer Machine, Data Flow Machine, rekursivní funkce, Markovovy algoritmy, reverzibilní algoritmy, buněčné a grafové automaty, ale třeba i dlaždičky v koupelně.

Operační systémy (*“Mám 3GHz procesor, tak co ty Windowsy už půl hodiny dělají?”*)

OS

Michal Vaner

Jak vypadá architektura dnešních operačních systémů aneb co musí programátor vědět, aby mu nepadala Wokýnka/Tučňáci. Správa procesů a vláken, plánování, synchronizace. Paměť, adresace a její přidělování. Správa souborů, filesystémy. Čemu se říká jádro a proč se spojuje s pudlem.

Filesystémy (*“Opravdu je FAT tabulka tlustá?”*)

FS

Jan Matějka

Povídání o tom, jak kdo ukládá data na disk. Dozvíte se, jak funguje filesystém FAT či jeho modifikace VFAT, jak ukládá data Linux na filesystémy EXT3, EXT4 či dosti netradiční ReiserFS. Nadějný nový BtrFS, který možná za pár let nahradí EXT3. Filesystémy pro SSD disky. NTFS. Jako perlička na závěr SpadFS.

Předpoklady: Hrubé povědomí o tom, jak fungují pevné disky.

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

UNIX

Martin Mareš, Jan Matějka

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.

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 postavit a jak v něm něco opravit. Kde najít nejnovější zdrojáky a kde najít pomoc, až se něco pokazí.

Spravujeme linuxový server (*“Kolik uživatelů ti nadává, tolikrát jsi adminem.”*)

SLS

Jan Matějka

Povídání o tom, co se všechno může stát, když spravujete server. Od uživatele, který prozradil heslo ke svému webmailu spambotovi, přes občasné čtení logů, zálohování apod. až do situace, kdy server spadnul a musíme vyrazit na místo. Užitečné pro začínající kořeny.

Předpoklady: Virtuální svetr do virtuální serverovny.

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

NET

Martin Mareš, Jan Matějka

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. Protokoly rodiny TCP/IP, síťové topologie (a proč Internet vlastně nemá žádou), internetworking. Pár taktů hudby budoucnosti: IPv6, multicasting, přenos v reálném čase atd.

Sítě II – aneb aplikační protokoly TCP/IP (*“Pokud jste se zamotali do sítí, tak se vás pokusíme vymotat.”*)

NET2

Martin Mareš, Jan Matějka

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

Jan Matějka, Martin Mareš

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.

E-mail (*“Drahoušek zákazník.”*)

EMAIL

Jan Matějka, 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 a BFU. Nakonec státní datové schránky a proč je to zlý ošklivý nepěkná věc.

Jednoúčelové mikroprocesory (*“Programujeme mikrovlnku, ledničku, bagr . . .”*)

MIKRO

Jan Matějka

Co ovládá vaše náramkové hodinky, MP3 přehrávač nebo třeba cyclocomputer. Ukážeme si, jak se programuje procesor bez OS. PIC kontra AVR. I2C, RS232, LCD displej, 7-segmentovka. Alarm a autoalarm, elektronika v autě a proč se nová auta opravují víc s notebookem než manuálně. Řízení světelné křižovatky dříve a nyní a proč není kruhový objezd lepší. Home-made řízení modelové železnice a srovnání s reálným provozem. Možná ukázky existujících zařízení.

Předpoklady: Základní znalost C nebo nějakého dialektu Assembleru

Cache oblivious algoritmy (*“Kešuješ, kešujem, 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 ukávek s poněkud překvapivým chováním.

Předpoklady: Kešu oříšky

Informatické přednášky – lingvistika a zpracování jazyků

Jazyky, gramatiky a automaty *

AUTO

Martin Mareš, Michal Vaner, Pavel Veselý

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?

Jazyková Zoo

JZOO

Martin Mareš, Jan Matějka

Programovací jazyky jsou všelijaké – procedurální, funkcionální či logické, typované silně, slabě nebo třeba i vůbec, objekt. . . stop, vykládat si o všelijakých rodech, družích a čeledích jazyků by byla nejspíš nuda, a tak si raději zajdeme do zoo a na ta zajímavější zvířátka se podíváme osobně: APL (či A^+ , případně J : průvan ve skladišti písmenek), Intercal (když existuje $GO\ TO$, proč by nemohlo existovat $COME\ FROM?$), Forth (pozpátku píšeme výrazy všechny úplně), Shakespeare (program coby divadelní hra), Ook!, Lingua::Romana::Perligata a další.

Kompilátory * (*“Jak se dělají kompilátory (a nebo komplikátory?)”*)

KOMP

Martin Mareš, Michal Vaner

Povídání o tom, jak překladače fungují uvnitř – jak se program parsuje, jak se optimalizuje kód atd. Co je to front end, back end, „middle end“, mezikód a jiná arkána umění kompilátorového. Jak psát programy tak, aby kompilátoru chutnaly, co optimalizovat ručně a co naopak udělá kompilátor lépe než my.

Předpoklady: Základní povědomí o tom, co to je procesor a co dělá.

Informatické přednášky – grafika a typografie

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

GFX

Martin Mareš, Pavel Čížek

Kreslení a zpracování obrazu na počítači. Souřadnice (rovinné, prostorové i barevné) a jejich transformace. Základní grafická primitiva: body, úsečky, kružnice, elipsy, Bézierovy křivky a jejich rasterizace. Vyplňování n -úhelníků a křivkou ohraničených oblastí, flood fill. Pár triků navíc: maticové filtry, anti-aliasing a dithering. Grafické formáty a komprese obrázků. Základy trojrozměrného promítání a vykreslování scény.

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

GEOM

Martin Mareš

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.

Barevné systémy (“*Co je na konci duhy?*”)

COLOR

Jan Matějka, Martin Mareš

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 (“*Vy obrázky malujete? To my je programujeme . . .*”)

PS

Martin Mareš, Jan Matějka

Jemný úvod do jazyka určeného k tisku grafiky a textu. Základní principy, řídicí konstrukce a datové struktury, cesty a kreslení objektů, transformace souřadnic, DSC komentáře. Co je to PDF (Portable Document Format). Různé druhy fontů (např. Type1, TrueType) a jak fungují.

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

MF

Jan Matějka

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, Jan Matějka

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

Martin Mareš, Jan Matějka

Z předchozí přednášky máme představu o tom, jak vypadá pěkná sazba. K její výrobě nám pomůže typografický systém \TeX . Praktická přednáška s ukázkami použití \TeX u od hladké sazby knihy až po zběsilosti hraničící s programováním. Jak do \TeX u vkládat obrázky a jak to raději nedělat. Kde shánět další informace: \TeX book, \TeX book naruby a další zajímavá literatura. Praktické rozdíly mezi různými dialekty \TeX u. Všelijaká rozšíření: pdf \TeX , e \TeX , Lua \TeX .

Matematické přednášky

Logika (“*Tato věta sem nepatří.*”)

LOGI

Martin Mareš, Pavel Čížek

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, Martin Böhm

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.

Martin Böhm

Někomu se může zdát dokazování vět na základě pravděpodobnostního argumentu jako čirá magie. Pokud tento sen skončí, cíl přednášky byl splněn. Použití pravděpodobnostní metody v důkazech existence kombinatorických objektů; metoda střední hodnoty, metoda malých změn, metoda druhého momentu, použití Lovászova lokálního lemmatu.

Předpoklady: Základy pravděpodobnosti (PP)

Teorie množin a matematika nekonečen * (“*Je Vlk nedosažitelný kardinál?*”)

TEMNO

Martin Mareš, Jan Matějka, Pavel Čížek

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. Ordinaly 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.

Grafy bez algoritmů

GRAFY

Martin Mareš, Martin Böhm, Lucie Mohelníková, Pavel Čížek

Teorie grafů trochu teoretičtěji. Různé druhy grafů a jejich vlastnosti. Vrcholové a hranové barvení grafů, Eulerova věta, hamiltonicity grafů, rovinné grafy a grafy na plochách, Kuratowského věta, Eulerova formule, věta o skóre, grafové minory.

Barevnost grafů * (“*Bílá, modrá, červená, co to pro graf znamená?*”)

BAGR

Martin Böhm

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. Barvení grafů na plochách vyššího rodu, channel assignment problem, hranová barevnost, listové barvení, vybíravost grafů a jejich tříd.

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

ROG

Martin Böhm, Martin Mareš, Pavel Veselý

Povídání o grafech, které jde nakreslit na papír bez křížení hran. O tom, 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. Proč je Carsten Thomassen ultra-geniální. Když zbyde čas, zkusíme grafy kreslit i na jiné plochy: kupříkladu Möbiovu pásku, pneumatiku nebo ušatou kouli.

Lineární algebra

LA

Martin Mareš, Michal Vaner, Lucie Mohelníková, Pavel Čížek

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.

Diskrétní optimalizace **

OPT

David Marek, Pavel Veselý, Lucie Mohelníková

Řešení úlohy lineárního programování. Simplexová metoda. Celočíselné programování, grafové problémy vyjádřené jako úlohy lineárního programování. Věta o dualitě. Totálně unimodulární matice. Párování, toky. Metoda řezů (cutting plane).

Předpoklady: LA

Matroidy ** (“*Co dostaneme křížením slona se šnekem?*”)

MATR

Martin Böhm

Teorie matroidů aneb jak to dopadne, když spojíme lineární algebru s teorií grafů a ještě do toho přisypeme hladové algoritmy. Co je to matroid, prostor cyklů, který matroid je reprezentovatelný a který grafový. Minory a miňonky, dualita a jiné ulity. Důkazy jednoduché, teorie fascinující.

Teorie (vesměs samoopravných) kódů (“*f y cn rd ths, y will b gd cmprtr 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, Pavel Čížek

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 quaterniony, octoniony a Cliffordovy algebry. Remember, life is complex.

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

NUT

Jan Matějka, Martin Mareš

Co a k čemu je teorie čísel. Počítání v kongruenci, Euklidův algoritmus a jeho použití. Malá Fermatova věta, Čínská zbytková věta a k čemu v praxi jsou. Jak si odvodit kritéria dělitelnosti. Malý výlet do algebry a příslušné zobecnění pár srandovních pozorování. Jak z toho všeho odvodit RSA – asi nepoužívanější asymetrický šifrovací algoritmus dnešní doby. Jak RSA funguje, proč funguje a jestli bude ještě fungovat. Generování klíčů, faktorizace kontra testování prvočíselnosti.

Fourierova transformace *

FFT

Martin Mareš, Pavel Čížek

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)

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.

Pokročilá kombinatorika (“Podlými podvody pokoutně připravíme předpis.”)

KOMB2

Jan Matějka

Vytvořující funkce coby velký podvod v mezích zákona. Výsledek příkladu si vycucáme z palce a pak dokážeme, že je správný. Malý výlet do algebraických končin a lemma, co není Burnsideovo.

Teorie kombinatorických her (“Život je jen hra . . . Jakou má vyhrávající strategii?”)

GAME

Martin Mareš, Pavel Veselý

Rozličné kombinatorické hry se zápalkami, kamínky, barvičkami či grafy. U některých si ukážeme výherní či obranné strategie, u některých dokážeme, že příslušná strategie existuje, i když nevíme, jak vypadá. Zmíníme například: všelijaké piškvorky, hex, různé varianty Nimu, vojáčky v poušti, speciality à la Herkules a Hydra, a další. Můžeme se zapovídat i o tom, jak podobné hry programovat na počítači.

Syntetická planimetrie (“Desátý bod kružnice devíti bodů”)

PLANI

Jan Matějka, Lucie Mohelníková

Klasická i šílená planimetrie podle přání publika. Konstrukce trojúhelníků ze všeho možného, Apolloniovy a Pappovy úlohy. Zobrazení od souměrností po afinitu a kruhovou inverzi. Geometrické důkazy a věty, různé středy trojúhelníka, Feuerbachova kružnice devíti bodů a další čtyři její významné body, Cevova a Menelaova věta. Kouzlo tětíkových čtyřúhelníků a nepřeberné množství dalších témat.

Předpoklady: Právítko a kružítko (alespoň virtuální)

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

DG

Jan Matějka

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.

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, kružítko a jeepu? Přímo se to dělá těžko, ale existuje spousta krásných triků, jak neřešitelnost problémů dokazovat. Nesložitelné hlavolamy, nerozvádětelné uzly, nepopsatelná čísla, neroztřetitelné úhly, nealgoritmické problémy a jiné slasti nekonstruktivní matematiky. Jak naopak ukázat, že něco existuje, aniž bychom věděli, jak to vypadá?

Derivace a integrály

DERIV

Pavel Čížek, Jan Matějka

Nejen ve fyzice se často setkáme s rovnicemi typu $a = \partial v / \partial t$. Jak se s tím počítá a proč nemůžeme zkrátit ∂ . Derivace, integrál, jak se počítají, a zbude-li čas, tak i některé obyčejné diferenciální rovnice.

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 hvězdopřevodci, „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.

Vesmír, aneb co se v něm (také) dá potkat (“A opravdu je černá díra černá?”)

ASTRO

Pavel Čížek

Trochu teoretičtější (a nezávislé) pokračování SKY. Co se děje s hvězdami na konci jejich života, co je to bílý trpaslík, neutronová hvězda, černá díra. Jaktože se černé díry dají pozorovat a proč jsou obvykle ve středu nejsilnějších pozorovaných zdrojů světla. Jak se vyvíjel vesmír, co je to reliktní záření, temná hmota, energie a proč současná teorie popisující velký třesk je asi v základech špatně. Obsah přednášky se upraví dle zájmu posluchačů (a mých znalostí).

Předpoklady: Fantazie a otevřená mysl. Jak se říká příroda je divočejší, než nejšilenější známá teorie.

Světlo ***LIGHT**

Pavel Čížek

To, že je světlo elektromagnetické vlnění, asi každý ví. To, že elektrická a magnetická složka ve světle svírají pravý úhel, je známo o něco méně. Ale jak se k tomu vlastně dospělo? Maxwellovy rovnice a jak z nich plyne světlo. Snellův zákon.

Předpoklady: Znat pojem derivace (DERIV), vektoru (LA) a obojí mít zažitě.

Newtonova mechanika, jak se nebere na střední škole ***NEWT**

Pavel Čížek

Fyzika na střední škole se vyhýbá pojmu derivace jako čert kříži. Co všechno se dá spočítat, když už víme, co to je? Keplerovy zákony, harmonický oscilátor, vrh vzhůru s brzděním vzduchu, prověšení drátu (jen ideálního ☺) atd.

Předpoklady: Vědět, co je derivace (DERIV) a nebát se jí.

Digitální elektronika a hradla**DIGI**

Martin Mareš, Jan Matějka

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.

Ostatní přednášky**Lingvištika** (*“Přísudek je v této větě podmět.”*)**LING**

Martin Mareš

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. Jak se jazyky vyvíjejí a jak se navzájem ovlivňují. Kde jsme přišli k pravidlům a jaký je jejich smysl. Existují synonyma? Proč je jazyk nejednoznačný a proč je to dobře. Jak se na jazyk dívá matematik a jak se na matematiku dívají lingvisté. Jak vzniklo písmo? A jak otazník? Jak zapsat zachrochtání a jak mlasknutí &c.

Lojban (*“Ženu holí stroj – kolik významů najdete?”*)**LOJB**

Martin Böhm

Už Vás nudí jazyky minulosti, které mají jinou výslovnost a jiný zápis? Obsahují výjimky z pravidel? Seznamy speciálních vyjmenovaných slov a interpunkci, která je jiná v úterý, v sobotu a ve větách hovořících o papoušcích? Vítejte do klubu! Povíme si něco o umělém jazyku, kteří byl vytvořen logiky pro logiky. A stroje mu skvěle rozumí!

MFF UK aneb co obnáší matfyzákem býti (*“Mamínko, 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.

Matfyzácké vtipy**MFV**

Karel Tesař

Hlavním tématem této přednášky bude rozbor matfyzáckých vtipů a polemizování nad nimi. Které vtipy jsou dokonale promyšlené a chápou je jen matfyzáci? A které naopak určitě matfyzáci nevymysleli? To a ještě více se dozvíte na této přednášce.

Orientace**ORI**

Martin Mareš

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.

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.

Počítače a paragrafy (*“Zavřete mě za znásilnění, mám nástroj.”*)**PRAVO**

Jan Matějka

Copyrighty, patenty, trademarky a podobný obtížný hmyz. V čem se liší a jak se vyhnout pobodání. K čemu jsou licence, oblíbené licence free softwaru a co dovolují. Jak škodí patenty, jak se jim vyhnout a kdy je lepší nevědět. Zbraně hromadného ničení v rukou velkých firem a co je patentový troll. Jak fungují trademarky a kdy se jim lze smát. Hudba, filmy, autorské svazy, poplatky, ACTA a co s tím má společného George Orwell. Přineste svoji oblíbenou EULA, zasmějeme se.

Předpoklady: Silný žaludek

Předmětové olympiády od A do Z (*“Byl jsem na deseti celostátních kolech. Kdo dá víc?”*)**SOUT**

Jan Matějka

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.

Efektní chemie (*“Zasyčí, nebo bouchne?”*)

CHEM

Jan Matějka

Tršakaviny a manipulace s nimi. Co je ještě bezpečné a co nevyrobět bez odborného dohledu. Proč něco vybuchuje i za vlhka a něco jen za vlhka. Nejsmradlavější látky světa, jak se dají vyrobit a proč to nedělat. Jak bezpečně vyrobit dvoumetrový kráter a jak spálit hliníkovou vidličku. Jak spočítat správný poměr surovin v zápalné směsi, jak (ne)odstřelit pařez. Které látky ve tmě svítí. Teoretická přednáška, žádné velké díry do světa (ani do stolu) nelze čekat, neboť letos nestíháme nakoupit suroviny.

Předpoklady: Nezáporný vztah k chemii

Železnice a kolejová doprava obecně (*“Vlak bude opožděn z důvodu ztráty lopatky na uhlí.”*)

RAIL

Jan Matějka

České koleje a co po nich jezdí, trocha historie. Život okolo železnice, řízení dopravy. Proč (asi) se srazily vlaky v Moravanech – metody zjišťování volnosti koleje a další speciality. Co je to šotouš a kde ho můžete spatřit. Vstup nových dopravců na české koleje, soudní spory mezi firmami v oboru a co to znamená pro naše železnice, lokomotiva řady 380 „Messerschmitt“.

Předpoklady: Alespoň letmé povědomí o tom, co je to železnice.

Křesťanství (*“A pane faráři, ukážete nám tu mučírnu, co máte ve sklepě?”*)

CHRIST

Jan Matějka

Existuje to už nějakých 2000 let a za tu dobu to infiltrovalo celou naši kulturu. Vývoj křesťanství od počátku do současnosti, proč už se neupalují čarodějnice a kdo to vlastně dělal. Jaké církve existují dnes a proč se nespojí. Mýty a předsudky, zastaralé informace a jak k tomu přispívá školní vyučování. Co to je mše, přijímání, posloupnost svatých apod. Problém financování církvi státem. Když zbyde čas, přijdou na řadu i vtipy a drby.

Mystika (*“Šaman, Buddha, Kristus, Mohamed i velký Tux jedno jsou.”*)

MYST

Jan Matějka

Povídání o tom, co mají společného tak zdánlivě odlišné filosofické směry (nebo náboženství) jako buddhismus, šamanismus, křesťanství, ... Nirvána jako cíl nebo prostředek. Přenos myšlenek na dálku. Psychosomatika a léčitelství, psychotropní látky, placebo efekt, homeopatika. Výchova, vůle, vedení a mnohé další. Povídání o temných koutech lidské duše.

Předpoklady: Ochota připustit existenci duše a iracionalitu člověka.

Mapování mysli a jiné techniky (*“Někdo mapuje terén, my mapujeme mysl”*)

MIND

Pavel Veselý

Jak zmapovat alespoň část svých myšlenek a jak se v mapě neztratit. Jak nejlépe mohou dosáhnout toho, aby mapovaná oblast při samotném mapování rostla. Využití map mysli při přípravě čehokoliv. Proč je velký čistý papír někdy lepší než počítač. Brainstorming, duševní rozcvičky, nasazování klobouků a další užitečné techniky, jak něco v krátkém čase vymyslet, a různé způsoby, kterak přijít na řešení problému.

Propagace a vnější vztahy (*“Dobrý den, můžeme si popovídat?”*)

PR

Jan Matějka

Máme úžasný produkt (službu, nabídku) a chceme mu udělat pořádnou reklamu. Propagace osobní i hromadná. Jak vyrobit dobrý plakát, jak hlásit do školního rozhlasu, co říct novinářům, aby výsledná zpráva aspoň trochu odpovídala realitě. Propagace zjevná i skrytá. Základní chyby, kterých se vyvarovat. Mýtická zkratka PR. Komerční propagace má jiná pravidla a metody, to je však již nad rámec této přednášky.

Úspěch a pracovní nasazení (*“Máš IQ větší než libovolná konstanta? To Ti stačit nebude. . .”*)

SUCC

Martin Böhm

Napůl přednáška, napůl diskuze o tom, co vlastně tvoří úspěch. Někteří tvrdí, že to je jen o vlastní vůli. Je to opravdu tak? Pravidlo 10 000 hodin. Dá se vůle naučit? Jak být doma co nejproduktivnější a jaký je skutečný smysl řešení seminářů. Talent a jestli je opravdu potřeba.

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. Těž o všelijakých substancích čaji podobných.

Harmonie (*“Uřčete zmenšenou sekundu od feses.”*)

HARM

Jan Matějka

V pozadí za poslouchanou hudbou se schovává pan režisér – harmonie. Proč se nám některé kombinace tónů poslouchají lépe a jiné hůře, jak je za sebe kombinovat. Co je to konsonance, disonance, personance a co s tím vším má společné Asonance. Harmonické konstrukce a postupy. Proč se pro různé hudební nástroje píše jiná hudba, proč zní stejný akord jinak na kytaru a na klavíru. Co je to čtyřakordák a proč má takový úspěch. Teorie proložená mnoha ukázkami; pokud zbyde čas, můžeme si složit třeba písničku, čtyřhlasý chorál nebo bachovskou dvouhlasou polyfonii.

Žonglování (*“Kaskáda, Fontána, Sprcha, Deštník, Větrný mlýn a spol.”*)

JUGGLE

Pavel Veselý

Velmi staré umění žonglování rozvíjí osobnost po mnoha stránkách, zlepšuje mimo jiné koordinaci, prostorovou orientaci a náladu. Jak začít a jak vytrvat. S čím vším se dá žonglovat, jaké všemožné triky existují a jak se zapisují pomocí čísel. Bonusem ukázky některých nepříliš těžkých triků. V případě zájmu možnost zapůjčení náčiní o poledních pauzách s radami pro začátečníky a mírně pokročilé (kdo umí s 5 a více míčky, může učit mě).

Michal Vaner

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?

Věda a akademická sféra (“Bez spolupráce nejsou vědecké koláče.”)

ACAD

Martin Böhm

Chtěli byste se na vysoké škole naučit řešit otevřené problémy, nebo třeba uvažujete o práci v akademické sféře? Budeme si povídat o tom, jak takové bádání probíhá: jaké problémy se obvykle řeší, jak taková spolupráce vypadá, kde se publikuje a tak podobně. Když bude zájem, můžeme otevřít debatu i o kontroverznějších tématech, které se kolem vědy točí.

Fotografie (“Kde najdu tlačítko »Krásná fotka«?”)

PHOTO

David Marek, Martin Böhm

Fotografování není jen o tom zamířit a zmáčknout spoušť. Představíme si druhy fotoaparátů, které jsou k dispozici. Dozvíme se o svaté trojici – clona, závěrka, citlivost. Jaké jsou rozdíly mezi objektivy a proč nebudou fotky z mého mobilu kvalitní, i když má 10 megapixelů? Co je to expozice? Po technické části se zaměříme na kompozici. Řekneme si, kterých pravidel je dobré se držet a kdy je můžeme porušit. Podíváme se na příklady jak se to má dělat a jak naopak ne.

SLOZ	Algoritmy a jejich složitost	1	CRYPT2	Kryptologie II	3
COLOR	Barevné systémy	8	LA	Lineární algebra	9
BAGR	Barevnost grafů	9	LING	Lingvistika	11
CACHE	Cache oblivious algoritmy	7	KERN	Linuxové jádro a jak se v něm vyznat	6
TEA	Čaj	12	LISP	LISP	4
CPP	Černá magie v C++	3	LOGP	Logické programování	4
CWIZ	C for wizards	3	LOGI	Logika	8
DS2	Datové struktury pro pokročilé	2	LOJB	Lojban	11
DS3	Datové struktury pro šílence	2	MAKE	Make	5
DS1	Datové struktury pro začátečníky	2	MIND	Mapování mysli a jiné techniky	12
DERIV	Derivace a integrály	10	MFV	Matfyzácké vtipy	11
DG	Deskriptivní geometrie	10	MATR	Matroidy	9
DIGI	Digitální elektronika a hradla	11	MF	MetaFont, MetaPost	8
OPT	Diskrétní optimalizace	9	MFF	MFF UK aneb co obnáší matfyzákem býti	11
DYNP	Dynamické programování	2	MODEL	Modely počítačů	6
PHP	Dynamický web a PHP	5	MYST	Mystika	12
CHEM	Efektní chemie	11	NZALG	Ne až tak základní algoritmy	1
EMAIL	E-mail	7	CESTY	Nejkratší a jiné cesty	1
EVA	Evoluční algoritmy	6	DNET	.NET Framework	3
FS	Filesystémy	6	NEWT	Newtonova mechanika, jak se nebere na střední škole	11
PHOTO	Fotografie	13	OBJ	Objektově orientované programování nejen v C++	3
FFT	Fourierova transformace	9	OTG	Omezené třídy grafů	2
FUNC	Funkcionální programování	4	OSM	OpenStreetMap	11
GDB	Gdb a jiné ladící nástroje	5	OS	Operační systémy	6
TEMPL	Generika	4	ORI	Orientace	11
GEOM	Geometrie a počítače	8	PARAL	Paralelní výpočty	2
GIT	Git a jiné systémy pro správu verzí	5	PERL	Perl	4
GA	Grafy & algoritmy	1	PRAVO	Počítače a paragrafy	11
GRAFY	Grafy bez algoritmů	9	GFX	Počítačová grafika	7
HARM	Harmonie	12	SKY	Podzimní obloha	10
HASK	Haskell	4	KOMB2	Pokročilá kombinatorika	10
HPC	High-Performance Computing	5	PYTH2	Pokročilé povídání o Pythonu	4
REGEX	Hledání v textu	2	PORT	Portabilní programování	5
ITREE	Intervalové stromy	2	PS	PostScript	8
STYLE	Jak se nestat vepřem	5	PS	Pravděpodobnost	8
SOL	Jak vypadá řešení	1	PP	Pravděpodobnost a algoritmy	2
JAVA	Java	3	PPALG	Pravděpodobnostní metoda	8
JZOO	Jazyková Zoo	7	PPMET	Předmětové olympiády od A do Z	11
SQL	Jazyk SQL	4	HW	Principy počítačů	6
XML	Jazyk XML a související technologie	4	THREAD	Procesy a vlákna	4
AUTO	Jazyky, gramatiky a automaty	7	GPU	Programování na grafické kartě	6
MIKRO	Jednoúčelové mikroprocesory	7	CSTR	Programování s omezujícími podmínkami	3
KOMB	Kombinatorika	10	PASM	Programování v assembleru	6
KOMP	Kompilátory	7	WIN	Programování ve Windows	5
CPLX	Komplexní a komplexnější čísla	9	C	Programování v jazyce C	3
PRESS	Kompresce dat	2	CIS	Programování v jazyce C#	3
CHRIST	Křesťanství	12			
CRYPT	Kryptologie	3			

PLX	Programování v Linuxu	5	KODY	Teorie (vesměs samoopravných) kódů	9
TEAM	Programování v týmu	12	TEX	T _E X	8
DFS	Prohledávání do hloubky	1	VIM	Textový editor Vim	5
GSEARCH	Prohledávání nejen grafů	1	TOKY	Toky v sítích	2
PR	Propagace a vnější vztahy	12	SLOZ3	Třídy složitosti	1
PYTH	Python	4	TYPO	Typografie	8
ROG	Rovinné grafy	9	UNIX	UNIX	6
NET	Sítě a Internet	6	SUCC	Úspěch a pracovní nasazení	12
NET2	Sítě II – aneb aplikační protokoly TCP/IP	7	RAMS	Úvod do Ramseyovy teorie	8
SLOZ2	Složitější složitost	1	NUT	Úvod do teorie čísel a RSA	9
MEM	Správa paměti	5	ACAD	Věda a akademická sféra	13
SLS	Spravujeme linuxový server	6	ASTRO	Vesmír, aneb co se v něm (také) dá potkat	10
STRG	Stringové algoritmy	2	VYCIS	Vyčíslitelnost	1
LYK	Stručný úvod do základů teorie vlkodlaků	1	HTTP	Web uvnitř	7
LIGHT	Světlo	10	ZALG	Základní algoritmy	1
PLANI	Syntetická planimetrie	10	RAIL	Železnice a kolejová doprava obecně	12
GAME	Teorie kombinatorických her	10	JUGGLE	Žonglování	12
TEMNO	Teorie množin a matematika nekonečen	9			
NONEX	Teorie nemožného	10			