

# PODZIMNÍ SOUSTŘEDĚNÍ KSP 2017 – 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 (např. 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

**Základní algoritmy a jejich složitost** (“*Čím menší je časová složitost algoritmu, tím větší je složitost kódu.*”) **ZAKL**

Pravděpodobně dvoudílná přednáška pro ty, kdo potřebují dohnat základní znalosti nutné pro ostatní přednášky. Zdefinujeme si základní pojmy jako je algoritmus, program, rekurze a jak se počítá jejich časová složitost, bude následovat přehled základních algoritmů – převážně třídění, rychlé hledání  $k$ -tého nejmenšího prvku, práce s výrazy a další.

**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**

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

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

Jirka Setnička, Martin Mareš, Dominik Smrž, Pěta Pelikánová

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

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

Jirka Setnička, Martin Mareš

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

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

Jirka Setnička, Filip Štědranský

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**

Kuba Maroušek, Filip Štědranský

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**

Jan Maria Matějka, Jirka Setnička, Kuba Maroušek, Filip Štědrnský, Pěta Pelikánová

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**

Kuba Maroušek, Filip Štědrnský, Martin Mareš

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.

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

**GEOM**

Jan Maria Matějka, Jirka Setnička

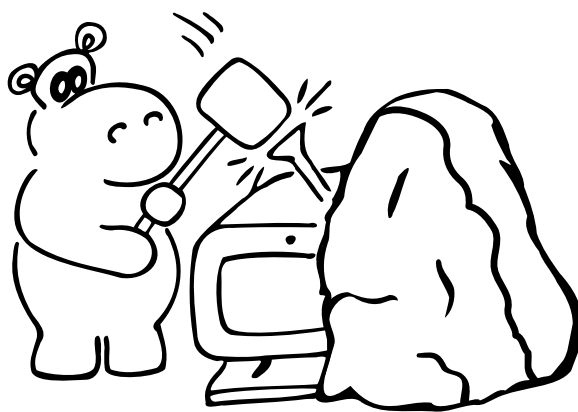
Základní algoritmy pro řešení geometrických úloh – konvexní obal, dva nejbližší body v rovině, výpočet obsahu nekonvexního mnohoúhelníka, lokalizace bodu, scanline algoritmus a jeho použití, Voroného diagramy a souvislost s persistentními datovými strukturami.

**Amortizace** (“Celek bývá daleko menší než součet částí.”)

**AMORT**

Martin Mareš

Spousta algoritmů je mnohem rychlejší, než jak na první pohled vypadají. Šikovný způsob, jak takové chování zkoumat, je amortizovaná časová složitost. Předvedeme několik trochu překvapivých příkladů amortizace: dvojková a jiná počítadla, datové struktury založené na přebudovávání, vyhledávací stromy bez otravného vyvažování, dynamizace datových struktur, udržování historie.



## Programovací jazyky a nástroje

**Programování v jazyce C**

**C**

Jan Maria Matějka, Jirka Setnička, Kuba Maroušek, Filip Štědrnský, Václav Končický

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

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

**OOP**

Kuba Maroušek, Filip Štědrnský, Dominik Smrž

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. Jak se liší OOP ve statických (C++, C#, Java) a dynamických (Python) jazycích. Jak programovat objektově i bez podpory jazyka, třeba v Céčku.

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

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

**PYTH**

Kuba Maroušek, Filip Štědrnský, Dominik Smrž

Jak programovat v Pythonu a jak v něm „nepsat Čeko“. 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.

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

**PERL**

Jan Maria Matějka, Kuba Maroušek, Pali Rohár

Jednoho dne se Larry Wall rozhodl, že nasype do jednoho velkého kotle spousty programovacích jazyků a unixových utilit, za stálého míchání povaří, posléze přecedí, přikoření a implementuje. Tak vznikl Perl, jazyk původně určený hlavně na zpracování textu, ovšem jak se ukázalo, též šikovný na spoustu dalších věcí. Asociativní pole, libovolně složitě 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. Co se Perl 5 přiučil od Perlu 6.

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

**THREAD**

Jirka Setnička, Filip Štědrský, Václav Končický, Pěta Pelikánová

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

*Předpoklady: Trochu představy o hardwaru*

**Logické programování** (“Mohu být svým vlastním dědečkem?”)

**LOGP**

Pali Rohár, Pěta Pelikánová

Což takhle projednou neříkat počítači, jak má věci počítat, ale jenom mu zadat podmínky, které má výsledek splňovat? Neprocedurální programování vychází přesně z této myšlenky. Podíváme se na programovací jazyk Prolog, který vychází z formální logiky. Zjistíme, které problémy se v něm neobvykle zjednoduší a které naopak programování promění v noční můru. Pokud jsi milovník rekurze, budeš u této přednášky nejspíš skoro celou dobu spokojeně vrnět.

**Haskell** (“V téhle proměnné je uložen okolní svět.”)

**HASK**

Kuba Maroušek, Václav Končický, Dominik Smrž, Pěta Pelikánová

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.*

**Programování webových aplikací (nejen) v Pythonu** (“Vlastní blog či wiki za 20 minut?”)

**WEBAPP**

Filip Štědrský

Obecná struktura webových aplikací a protokolu HTTP, v případě potřeby základy HTML. Jak jednoduše vytvářet webové aplikace v Pythonu pomocí webových (mirko)frameworků (Flask, Django, ...): zpracování požadavků a dat z formulářů, generování HTML, práce s databází. Dynamické stránky: AJAX, Server-Sent Events a další. Co nejde napsat na pár řádek, nás nezajímá.

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

**SQL**

Kuba Maroušek, Václav Končický

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 triggerry. Rozdíly mezi dialekty SQL.

## Hardware a operační systémy

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

**HW**

Jirka Setnička, Filip Štědrský, Martin Mareš

Vydáme se do země skřítků, kteří pohánějí počítače. Počítačové architektury od hodiniek 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.

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

**KOMP**

Filip Štědrský, Martin Mareš, Pěta Pelikánová

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. Pojdme zjistit, co jsou zač a co všechno s programem provádějí. Co udělá kompilátor za nás a co musíme naopak udělat my za něj.

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

**UNIX**

Jirka Setnička, Filip Štědrský

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

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

**SHELL**

Jirka Setnička, Kuba Maroušek, Filip Štědrský

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

**Programování v Linuxu** (“Všechno na světě je tak trochu soubor”)

**PLX**

*Jan Maria Matějka, Pali Rohár, Filip Štědronský, Martin Mareš*

Jak vypadá rozhraní mezi jádrem Linux a uživatelskými programy. Co se doopravdy stane, pokud ve svém céčkovém programu zavoláme `printf` nebo `malloc`. Jak napsat program, který vůbec nepotřebuje standardní céčkovou knihovnu. Co všechno se umí chovat jako soubor a co jako signál.

*Předpoklady: Schopnost přečíst a napsat jednoduchý program v C.*

## Sítě a bezpečnost

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

**NET**

*Jirka Setnička, Kuba Maroušek, Pěta Pelikánová*

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

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

**NET2**

*Jirka Setnička, Kuba Maroušek*

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

*Předpoklady: Základní povědomí o počítačových sítích*

**Kryptografie** (“`Gbgb arav zbp gnwan mcenin`.”)

**CRYPT**

*Filip Štědronský, Martin Mareš*

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

## Teoretická informatika

**Složitější složitost \***

**SLOZ**

*Kuba Tětek*

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 redukci, problémy NP-úplné a ještě těžší.

*Předpoklady: SLOZ*

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

**EVA**

*Jirka Setnička*

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

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

**MACHINE**

*Kuba Maroušek*

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

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

**MODEL**

*Jirka Setnička, Martin Mareš*

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. Trocha minimalismu aneb stroj s počítadly. Až nám začne být smutno, pořídíme si klidně  $N^2$  procesorů a spráhneme je do paralelního počítače (PRAM). Rychlé paralelní slévání a třídění. Pokud zbude čas, ukážeme si buněčné a grafové automaty, nebo třeba `dlaždičky v koupelně`.

**Jazyky, gramatiky a automaty \***

**AUTO**

*Kuba Tětek, Václav Končický*

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

*Kuba Tětek, Václav Končický, Dominik Smrž, Pěta Pelikánová*

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

## Úvod do teorie čísel

NUT

*Martin Mareš, Jan Maria Matějka*

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

*Kuba Tětek*

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.

## Funkce a jejich chování

MA

*Pěta Pelikánová*

Profrčíme letem světem základní tvary funkcí. Vrhne se na derivování a integrování. Podíváme se na aplikace při popisu chování funkcí a počítání rozměrů ohraničených ploch. Pak to všechno dáme do pohybu a budeme schopni určit objem rotačních těles.

## Jak vypadá zrcadlo v číslech

LIN1

*Pěta Pelikánová*

Na přednášce se podíváme na to, jaká matematika stojí za grafickými transformacemi obrázků. Ukážeme si, jak můžeme otáčet předmět pomocí jeho zobrazení osovou souměrností, a budeme si hrát s dalšími transformacemi. Dozvíte se, jak vypadají matice a jak je násobit s vektory. Povíme si, co všechno stojí na lineární algebře a jaké problémy dokáže vyřešit.

## Lineární algebra \*

LIN2

*Pěta Pelikánová*

Naučíme se hledat řešení soustav rovnic na papíře tak, abychom toho museli napsat co nejméně (Gaussova eliminace). Zabředneme hlouběji do vektorových prostorů a budeme hledat jejich hezké ortogonální a ortonormální báze.

*Předpoklady: Je dobré vědět, co je to matice a jak se násobí (přednáška LIN1).*

# Rozšiřující přednášky

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

## Algoritmy a datové struktury

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

**CESTY**

*Jirka Setnička, Filip Štědranský*

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.

**Datové struktury pro ještě pokročilejší \*\*** (“log log log log ... glo glo glo ...”)

**DS3**

*Kuba Tětek, Martin Mareš*

Na přednášce si ukážeme některou z méně známých složitějších datových struktur. Pokud Ti ostatní přednášky přijdou moc jednoduché, tato je ta pravá pro Tebe.

**Suffixové stromy \*\*** (“Jak obrátit řetězec naruby?”)

**SUFF**

*Filip Štědranský, Martin Mareš*

Suffixový strom je zajímavá datová struktura, pomocí níž jde vyřešit většinu řetězcových problémů v lineárním čase. Podíváme se, jak suffixový strom vypadá, k čemu se hodí a jak ho sestavit. Též prozkoumáme několik příbuzných zvířátek, jako třeba suffixové pole a suffixový automat.

**Stromové algoritmy** (“Půjdeme na to od lesa”)

**TREES**

*Martin Mareš*

Stromy jsou jednou z nejtýpčtějších (a nejjednodušších) odrůd grafů. Ledacos pro ně umíme řešit mnohem rychleji než pro obecné grafy, tak se pojdme podívat, jak se to dělá. Předvedeme několik obecných technik pro práci se stromy: DFS očíslování, „vandalskou indukci“, intervalové reprezentace. Různé rozklady: heavy-light, Fredericksonův, separátorový a ST-stromy.

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

**MAGIC**

*Martin Mareš, (Jenda Hadrava)*

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 příhrádkového třídění. Bitové kejklřství. Hledání největší díry.

## Programovací jazyky

**C for wizards \*** (“1 [x] +++++x [1]”)

**CWIZ**

*Jan Maria Matějka, Martin Mareš*

Ponořme se do hlubin Cčka, 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 Cčka od K&R až po nejnovější normu C11 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.*

**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.

**Jazyk Go**

**GOLANG**

*Jirka Setnička*

Go je moderní kompilovaný jazyk (vyvinutý původně v Google), který se pokouší být takovým Cčkem na steroidech. Umí být podobně rychlý, zaručuje větší typovou bezpečnost, ale má i prvky dynamicky typovaných jazyků. Jeho velkou silou je velmi snadné provázání na existující kód v C/C++, systém balíčků distribuovaných převážně přes Github, funkce vracející libovolný počet hodnot nebo třeba vestavěná podpora Unicode a vestavěná hashovací tabulka. Také se silně dbá na coding-style pro zajištění snadné čitelnosti programů.

**Jazyk Lua** (“Co není potřeba, to zahodíme.”)

**LUA**

*Jan Maria Matějka*

Předvedeme si minimalistický jazyk, jehož interpret se vejde do jednoho megabajtu, takže je horkým kandidátem na skriptovací jazyk pro zařízení, která mají málo paměti. Naučíme se nejen, jak takový jazyk používat, ale také, jakým způsobem jej propojit s jinými jazyky, především s jazykem C, a k čemu to všechno může být dobré.

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

**PERL6**

Filip Štědronský, 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, Dominik Smrž

Povídání o méně zmiňovaných částech Pythonu. Dekorátory, metaclassy, generátory, funkcionální styl programování v Pythonu. Jak napsat quicksort jako lambda funkci. Představení zajímavých modulů nejen ze standardní knihovny. Další témata dle přání účastníků: paralelní programování (asyncio, multiprocessing), síťová komunikace, GUI, matematické výpočty, propojení Pythonu s C, ...

Předpoklady: PYTH

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

**CPP**

Filip Štědronský

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

Předpoklady: základní znalost C++, staticky alokovaný kyblík

## Programovací nástroje a techniky

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

**GIT**

Jan Maria Matějka, Jirka Setnička, Pali Rohár, Filip Štědronský, Václav Končický

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.

**Git pro pokročilé** (“In case of fire, commit, push, and exit the building.”)

**GIT2**

Martin Mareš, Pali Rohár

Používáte Git pro všechny své programy a k svačině místo novin čtete commit logy svých oblíbených projektů? V tom případě pojďme nahlédnout pod pokličku, jak Git funguje uvnitř. Reprezentace historie pomocí hešování grafů. Pracovní strom, index, commity a jejich adresy, větve. Pack files jako elegantní způsob komprese dat na disku i na síti. Kouzelnické triky: hledáme bugy půlením historie, přepisujeme dějiny, automaticky konvertujeme soubory. Git v praxi: jak se liší správa zdrojů v projektech o jednom, deseti a tisíci programátorech. Udržujeme patche k cizímu programu aneb StGit.

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

**MAKE**

Jan Maria Matějka, 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 tomu nepomáhají ani automake, cmake, qmake a další.

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

**GDB**

Jan Maria Matějka, Pali Rohár, Filip Štědronský

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 (odšívovač), 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**

Jan Maria Matějka, Filip Štědronský, Martin Mareš

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

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

**STYLE**

Martin Mareš

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 vybdnout.

**Kompresie dat** (*“Jnm idln kpln j nstlčtln.”*)

**PRESS**

*Jirka Setnička, Filip Štědranský, 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).

## Hardware a operační systémy

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

**CACHE**

*Jirka Setnička, Kuba Tětek, Martin Mareš*

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*

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

**FS**

*Pali Rohár, Filip Štědranský, Václav Končický*

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 filesystemy od EXT2 k EXT4. Nadějný nový BtrFS, který je možná za pár let nahradí. Co se hodí na SSD.

## Sítě a bezpečnost

**E-mail** (*“Drahoušek zákazník.”*)

**EMAIL**

*Pali Rohár*

Co se stane s e-mailem, když jej odešlete? Kudy chodí a kudy jej čerti nesou? Jaké máte záruky, že přijde; proč občas přijde pozdě nebo vůbec. Problém formátů a kódování, chyby webových i jiných klientů. Protokoly SMTP, POP, IMAP a co se stane, když do nich přimícháme SSL/TLS. E-mailová bezpečnost, SPAM a (nefunkční?) obrana pomocí SPF, DKIM a DMARC. Nakonec se podíváme na ne zrovna triviální grafový problém, který je v emailech skrytý.

**IPv6** (*“Viac adries znamená aj viac problémov”*)

**IPV6**

*Pali Rohár*

Internetoví provideri nám pomalu začínají ponoukat připojení do světa IPv6 Internetu. Čo to IPv6 je? Čím sa líši od svojho staršieho kolegu IPv4? Prečo by sa o neho mal zaujímať aj obyčajný užívateľ? Pozrieme sa aké problémy IPv6 rieši, aké má výhody oproti IPv4 ako aj na súčasný stav podpory IPv6 v operačných systémoch. A hlavne s akými problémami sa človek bežne stretne než si bude môcť doma spraviť IPv6 sieť.

*Předpoklady: NET*

**Bezpečné programovanie** (*“V každom programe je aspoň jedna bezpečnostná chyba, tak poďme to napraviť”*)

**SECPRG**

*Pali Rohár*

Dost často sa programy nepíšu iba na jedno spustenie, ale sa používajú dlhé roky. Veľa z nich naviac bude používať úplne niekto iný. Na tejto prednáške sa pozrieme na časté chyby v programoch a ako im predchádzať. Ako klasifikovať či chyba je závažná a bezpečnostná. Ďalej si povieme, čo robiť v prípade, ak nájdeme nejakú (možno bezpečnostnú?) chybu. Komu ju oznámim a komu radšej nie.

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

**CRYPT2**

*Filip Štědranský, Martin Mareš*

Pokročilejší a občas nečekané aplikace základních kryptografických primitiv. Jak přesvědčit server, že známe heslo, aniž bychom mu ho posílali? Jak zajistit, aby útočník nemohl dešifrovat komunikaci, ani když dodatečně získá soukromý klíč? Jak funguje BitCoin (decentralizovaná digitální měna) či Tor (protokol znemožňující komukoli po cestě vědět, kdo s kým komunikuje)?

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

**Praktická kryptografie** (*“A proč jsou všechny ty zámky na papírových dveřích?”*)

**PCRYPT**

*Filip Štědranský, Martin Mareš*

Programátoři si často myslí, že pro bezpečnou komunikaci stačí vybrat si z knihovny osvědčenou silnou šifru. Jak naivní! Navrhnout bezpečný protokol není maličkost a dá se při tom ledacos zpackat. Replay útoky (jak otevřít auto krabičkou za 30 dolarů), útoky na padding a na blokovou strukturu. Či že je ten podpis? Jak nepoužívat RSA a jak nehešovat hesla. Jak náhodná jsou vaše čísla? Postranní kanály: časování, spotřeba, záření. K čemu se crackerům hodí termoska s tekutým dusíkem.

## Grafika a typografie



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

**COLOR**

*Jan Maria 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 neduh. „Systém“ Pantone. Reálné kontra imaginární barvy aneb proč nejde vyfotit duha.

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

**TYPO**

*Jan Maria Matějka, Martin Mareš*

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 T<sub>E</sub>Xnician.*”)

**TEX**

*Jan Maria Matějka, Jirka Setnička, Václav Končický*

Z předchozí přednášky máme představu o tom, jak vypadá pěkná sazba. K její výrobě nám pomůže typografický systém T<sub>E</sub>X. Praktická přednáška s ukázkami použití T<sub>E</sub>Xu od hladké sazby knihy až po zběsilosti hraničící s programováním. Jak do T<sub>E</sub>Xu vkládat obrázky a jak to raději nedělat. Kde shánět další informace: T<sub>E</sub>Xbook, T<sub>E</sub>Xbook naruby a další zajímavá literatura. Praktické rozdíly mezi různými dialekty T<sub>E</sub>Xu. Všelijaká rozšíření: pdfT<sub>E</sub>X, eT<sub>E</sub>X, LuaT<sub>E</sub>X.

**Formát PDF**

**PDF**

*Jan Maria Matějka, Martin Mareš*

Jeden z nejrozšířenějších formátů na předávání dokumentů má za sebou spletitou historii i dokumentaci. Ukážeme si, jak vypadá uvnitř a co se do něj dá uložit: grafické objekty, text, fonty, odkazy, všelijaké anotace a meta-data, a dokonce i kryptografické podpisy. Zmíníme se o profilech, třeba PDF/X a PDF/A. Při troše štěstí si vytvoříme jednoduchý PDF soubor ručně a možná půjde i otevřít.

## Teoretická informatika

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

**CSTR**

*Jirka Setnička*

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.

**Persistentní datové struktury** (“*Datové struktury cestující časem*”)

**PERS**

*Kuba Tětek, Martin Mareš*

Ukážeme si (téměř) obecný způsob, jak naučit datové struktury zapamatovat si celou svou historii. Předvedeme si, jak tuto historii modifikovat a k čemu to je celé dobré.

**Třídy složitosti \*** (“*Kolik sekund stojí jeden bajt?*”)

**SLOZ3**

*Martin Mareš*

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*

## Aplikace informatiky

**Hudba pro programátory** (“*Máte správně naladěnou klávesnici?*”)

**MFP**

*Jan Maria Matějka*

Počítače dnes dokáží dělat s hudbou divy. Ukážeme si analýzu nahraného zvuku i automatický generátor kytarových akordů. Nahlédneme pod ruce skladatelům herní hudby a vysvětlíme si, jak může hudba přirozeně přejít do jiného stylu v libovolné herní situaci. To vše na základě fyzikálních principů a hudební teorie.

**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, bílé křížky na asfaltu, ...

## Matematické přednášky

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

**DG**

*Jan Maria Matějka*

Trojrozměrný svět se dá hezky splácnout do dvourozměrného. Ukážeme si, jak narysovat trojrozměrné těleso na papír, jak si ručně pořídit různé průniky nebo sjednocení. Výstupem celé přednášky bude vystřižovánka nějakého složitějšího geometrického tělesa.

*Předpoklady: Prostorová představivost výhodou, pravítka a kružítko rovněž, koulítko a rovinítko netřeba.*

## Pravděpodobnost

PAST

Kuba Tětek, Václav Končický

Jak pracovat s pravděpodobností matematicky. Ukážeme si pravděpodobnosti jevů, nezávislé jevy střední hodnotu, náhodné proměnné a další. Také si vše procvičíme na několika příkladech. Pokud zbyde čas, tak si také ukážeme, jak se dá pravděpodobnost využít v informatice.

**Teorie množin a matematika nekonečen \*** (*“Kdo je nejvyšším z kardinálů?”*)

TEMNO

Martin Mareš

Historie matematiky je dlážděna trampoty s nekonečnem. Začalo to roztomilým problémem s želvou pana Zénona a vedlo až k poněkud děsivým paradoxům 18. století. V moderní době jsme se proti tomu obrnila teorií množin, na níž je dnes takřka celá matematika postavena. Jak se taková teorie buduje a jak se pomocí ní popisují nekonečné objekty. Množiny a jejich velikosti. Cantorův diagonální trik. Ordinály a houšť kardinálů. Potenciální kontra aktuální nekonečno. Jak si přiřadit přirozená čísla a jak ta reálná. Potíže s axiomem výběru.

**Teorie čísel a RSA \*** (*“ $2^{67} - 1 = 193\,707\,721 \cdot 761\,838\,257\,287$ ”*)

NUT2

Martin Mareš

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.

**Fourierova transformace \*\***

FFT

Martin Mareš, Jenda Hadrava

Jak rychle umíte násobit  $n$ -ciferná čísla? My to umíme lineárně. Hodí se k tomu chytrý trik pana Fouriera, který už dávno patří k matematické a fyzikální klasice. Ukážeme, co je Fourierova transformace zač, jak ji rychle spočítat a k čemu je dobrá: rychlé násobení polynomů i čísel, digitální zpracování zvuku a obrazu (spektrální analýza či třeba komprese).

*Předpoklady: Základy komplexních čísel*

**Meta-matematika** (*“Tato věta sem nepatří.”*)

METAM

Martin Mareš

Pokud budeme v životě věřit všemu, co je „přeci zřejmé“, dostaneme se brzy do potíží a v matematice to platí dvojnásob. Přírodní vědy si vymyslely opakovatelné pokusy a matematici axiomatický přístup. Ukážeme, jak z jednoduché sady axiomů vybudovat takřka celou matematiku. Dokonce tak, že správnost důkazů za nás ověří počítač, aspoň když mu trochu pomůžeme. Nadšení trochu ochladí Gödelova věta: ať děláme, co děláme, vždy zbude nějaké nerozhodnutelné tvrzení. Pomůže přidávat axiomy? Asi ne, ale za odměnu získáme mnoho různých matematik. A dá-li bůh, stihneme dokázat jeho existenci i neexistenci ☺.

**Intervalové počítání** (*“Prima, sekunda, tercie, . . . nekonečno.”*)

INT

Péťa Pelikánová

Co s čísly, která si počítač nezapamatuje přesně? Vezmeme si na pomoc intervaly a zabalíme je do nich. Naučíme se s intervaly počítat a nakrmíme nimi i funkce. Řekneme si, kde se dají využít v praxi, ale i v teoretické informatice.

**Teorie (vesměs samoopravných) kódů** (*“f y cn rd ths, y wll b gd cmpr prgrmmr!”*)

KODY

Martin Mareš

Jak komunikovat po lince, která průměrně každý  $k$ -tý bit přenesne 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?

# 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.

**Efektní chemie** (*“Zasyčící, nebo bouchne? A zavřou mě za to?”*)

CHEM

Jan Maria Matějka

Jak si pořídit díru do ruky efektně a jak hloupě. Proč jsou komerční rachejtle nudné a proč je lepší si dát pozor i na ně. Jak si pořídit kráter a jaké jsou právní důsledky takového jednání. Které látky ve tmě svítí a co se s nimi dá dělat. Částečně právnická, částečně chemická přednáška. Při troše štěstí bude i pokus.

**Předmětové olympiády od A do Z**

SOUT

Jan Maria 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.

**Hippo:Web** (*“Znovu vynalézáme kolo, ať konečně není hranaté.”*)

HIPPO

Jan Maria Matějka, Martin Mareš

Organizátoři KSPčka rádi dělají pokusy na sobě samých, a tak při vývoji nového webu semináře pohrdli všemi vyzkoušenými webovými frameworky a napsali si potvrzku jménem Hippo. Trochu netradiční, ale v mnoha ohledech velice příjemnou. Namátkově: HTML coby datová struktura, triky s kryptografickými tokeny, kombinování programovacích jazyků a tvorba dynamických stránek, které jsou přitom skvěle cacheovatelné. Případně také nahlédnutí do dalšího zákulisí KSPčka.

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

ORG

Jirka Setnička

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

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

ZDRAV

Jirka Setnička

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.

**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.

**Lockpicking** (*“Jak si odemknout, když si náhodou my (nebo soused) zapomeneme klíč :-)”*)

PICK

Jirka Setnička

Jak fungují dnešní zámky, co jsou to stavítka a jak vlastně fungují klíče. A jak se pomocí jednoduchých nástrojů dají využít výrobní nedokonalosti zámků k jejich odemčení. Použití planžet, napínáků, praktické ukázky odemykání, nastínění technik bumpingu a dalších postupů, jak se dostat přes zamčené dveře.

**Létání a natáčení s dronem** (*“Ne ne, to není řízená raketa, tam jen visí velký foťák.”*)

DRON

Jirka Setnička

Na co je dobré mít drona, kdy se dá reálně použít a kdy je to jen drahá hračka. Přehled o technologiích, jejich možnosti a omezení. Jak se liší armádní drony od dronů pro kameramany a nebo od dronů, které pořídíte za pár šupů v hračkářství. Také lehký výlet do legislativy – s čím smíme létat, kde smíme létat a co s tím můžeme dělat. Pokud počasí dá, tak i praktické hrátky s dronem.

**Biologická 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.*

Filip Štědranský

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

**Shell pro shíllence**

SS

Filip Štědranský

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

**Racionalita** (*“Pokud je tvrzení X pravdivé, chci věřit, že je pravdivé. Pokud ne, chci věřit, že je nepravdivé.”*)

RAT

Filip Štědranský

Naše přesvědčení o světě jsou jako mapa nějakého území. Měli bychom se snažit, aby mapa co nejlépe odrážela skutečnost, neustále ji upravovat podle nově objevených faktů. Přesto se lidi často spíš snaží přesvědčit ostatní i sebe, že jejich názor je správný, než aby věnovali veškeré úsilí nalezení názoru nejlépe kopírujícího realitu. Odhlížíme od nepříjemných pravd. Proč? Jak se naučit své názory absolutně podřídit realitě, přijímat svět, jaký je? Jaké systematické chyby v uvažování lidský mozek dělá a jak nám jejich znalost (ne)může pomoci? K čemu se hodí znát teorii pravděpodobnosti či úvahy z oblasti umělé inteligence?

**Naturalismus** (*“Tuto hypotézu jsem nikde nepotřeboval.”*)

NATUR

Filip Štědranský

Naturalismus je teorie, že svět řídí neměnné přírodní zákony, které mají poměrně jednoduchý matematický charakter, jsou celkem lhostejné k lidským osudům, měří stejně dobrým i zlým. Že život není magická vitální esence, nýbrž sada komplexních chemických interakcí. Že myšlení je výpočetní proces jako každý jiný. Naturalismus není axiomem vědy, nýbrž jedním z jejích největších objevů. Proč si dnes myslíme, že je nejspíš pravdivý? Jak vypadala mnohasetletá bolestivá cesta k němu a proč se z ní svým způsobem vzpamatováváme dodnes? Jaký vliv to má na naše vnímání hodnot, života a jeho smyslu?

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

YPLS

Martin Mareš &amp; Filip Štědranský

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

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

LING

Martin Mareš

Převážně nevážné a mírně nepřed-vidatelné po-vidání 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.

**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 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.

**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.

**Vektorizace obrázků** (*“Jak se rodí hroši?”*)

SVG

Péťa Pelikánová

Každý hezký text se neobejde bez obrázků. Dozvíte se jak se z hlavy někoho kreativního dostane obrázek hrocha až na leták KSP. Jak a proč z náčrtku tužkou udělat vektorový obrázek a v čem se dá tvořit vektorová grafika.

Pěta Pelikánová

Kde je hranice sportem k u zdraví či k trvalé invaliditě? Různé pohledy na cirkus. „Wow, to je nemožné!“ vs. „Jé, jak to dělají?“ Co je to nový cirkus, akrojóga, handstandy a headstandy ... Ať už si chceš o tom něco poslechnout nebo si chceš něco z toho vyzkoušet, ulov si Pětu.

**Internet of Things**

IOT

Pěta Pelikánová

Nahlédneme co vše se skrývá pod pojmem Internet of Things. Jak internet propojuje spoustu různých zařízení a jak se z nich shromažďují informace. Představíme si model klaudových služeb. Jak může vypadat IoT chytrá domácnost.

## Abecední seznam přednášek

**LYK** Stručný úvod do základů teorie vlkodlaků.. 1

### Základní přednášky

<b>AMORT</b>	Amortizace .....	2	<b>KOMP</b>	Od zdrojáku k programu .....	3
<b>DS2</b>	Datové struktury pro pokročilé .....	1	<b>PARSE</b>	Parsing čili analýza textu .....	2
<b>DS1</b>	Datové struktury pro začátečníky .....	1	<b>PERL</b>	Perl .....	2
<b>DYNP</b>	Dynamické programování .....	1	<b>HW</b>	Principy počítačů .....	3
<b>EVA</b>	Evoluční algoritmy .....	4	<b>THREAD</b>	Procesy a vlákna .....	3
<b>MA</b>	Funkce a jejich chování .....	5	<b>C</b>	Programování v jazyce C .....	2
<b>GEOM</b>	Geometrie a počítače .....	2	<b>PLX</b>	Programování v Linuxu .....	4
<b>GA</b>	Grafy & algoritmy .....	1	<b>WEBAPP</b>	Programování webových aplikací (nejen) v Pythonu .....	3
<b>GRAFY</b>	Grafy bez algoritmů .....	5	<b>DFS</b>	Prohledávání do hloubky .....	1
<b>HASK</b>	Haskell .....	3	<b>PYTH</b>	Python .....	2
<b>REGEX</b>	Hledání v textu .....	2	<b>NET</b>	Sítě a Internet .....	4
<b>ITREE</b>	Intervalové stromy .....	1	<b>NET2</b>	Sítě II – protokoly a síťové útoky .....	4
<b>LIN1</b>	Jak vypadá zrcadlo v číslech .....	5	<b>SHELL</b>	Skriptování v shellu .....	3
<b>SQL</b>	Jazyk SQL .....	3	<b>SLOZ2</b>	Složitější složitost .....	4
<b>AUTO</b>	Jazyky, gramatiky a automaty .....	4	<b>MACHINE</b>	Strojové učení .....	4
<b>KOMB</b>	Kombinatorika .....	5	<b>TOKY</b>	Toky v sítích .....	1
<b>CRYPT</b>	Kryptografie .....	4	<b>UNIX</b>	UNIX .....	3
<b>LIN2</b>	Lineární algebra .....	5	<b>NUT</b>	Úvod do teorie čísel .....	5
<b>LOGP</b>	Logické programování .....	3	<b>ZAKL</b>	Základní algoritmy a jejich složitost .....	1
<b>MODEL</b>	Modely počítačů .....	4			
<b>OOP</b>	Objektově orientované programování .....	2			

### Rozšiřující přednášky

<b>CRYPT2</b>	Aplikace kryptografie .....	8	<b>PDF</b>	Formát PDF .....	9
<b>COLOR</b>	Barevné systémy .....	9	<b>FFT</b>	Fourierova transformace .....	10
<b>SECPRG</b>	Bezpečné programování .....	8	<b>GDB</b>	Gdb a jiné ladící nástroje .....	7
<b>CACHE</b>	Cache-oblivious algoritmy .....	8	<b>GIT</b>	Git a jiné systémy pro správu verzí .....	7
<b>CWIZ</b>	C for wizards .....	6	<b>GIT2</b>	Git pro pokročilé .....	7
<b>BAR</b>	Čárové kódy .....	9	<b>MFP</b>	Hudba pro programátory .....	9
<b>CPP</b>	Černá magie v C++ .....	7	<b>INT</b>	Intervalové počítání .....	10
<b>DS3</b>	Datové struktury pro ještě pokročilejší .....	6	<b>IPV6</b>	IPv6 .....	8
<b>DG</b>	Deskriptivní geometrie .....	9	<b>STYLE</b>	Jak se nestat vepřem .....	7
<b>EMAIL</b>	E-mail .....	8	<b>GOLANG</b>	Jazyk Go .....	6
<b>FS</b>	Filesystémy .....	8	<b>LUA</b>	Jazyk Lua .....	6

<b>PRESS</b>	Komprese dat . . . . .	8	<b>PASM</b>	Programování v assembleru . . . . .	6
<b>MAGIC</b>	Magické algoritmy . . . . .	6	<b>TREES</b>	Stromové algoritmy . . . . .	6
<b>MAKE</b>	Make . . . . .	7	<b>SUFF</b>	Suffixové stromy . . . . .	6
<b>METAM</b>	Meta-matematika . . . . .	10	<b>NUT2</b>	Teorie čísel a RSA . . . . .	10
<b>CESTY</b>	Nejkratší a jiné cesty . . . . .	6	<b>TEMNO</b>	Teorie množin a matematika nekonečen . . . . .	10
<b>PERL6</b>	Perl 6 . . . . .	7	<b>KODY</b>	Teorie (vesměs samoopravných) kódů . . . . .	10
<b>PERS</b>	Persistentní datové struktury . . . . .	9	<b>TEX</b>	TeX . . . . .	9
<b>PYTH2</b>	Pokročilé povídání o Pythonu . . . . .	7	<b>VIM</b>	Textový editor Vim . . . . .	7
<b>PCRYPT</b>	Praktická kryptografie . . . . .	8	<b>SLOZ3</b>	Třídy složitosti . . . . .	9
<b>PAST</b>	Pravděpodobnost . . . . .	10	<b>TYPO</b>	Typografie . . . . .	9
<b>CSTR</b>	Programování s omezujícími podmínkami . . . . .	9			

## Půlnoční přednášky

<b>ACRO</b>	Akrobacie aneb létání . . . . .		<b>PICK</b>	Lockpicking . . . . .	11
<b>CAR</b>	Auto z pohledu technika . . . . .	11	<b>NATUR</b>	Naturalismus . . . . .	12
<b>EVOL</b>	Biologická evoluce . . . . .	11	<b>ORG</b>	Organizování . . . . .	11
<b>TEA</b>	Čaj . . . . .	12	<b>ORI</b>	Orientace . . . . .	12
<b>CHEM</b>	Efektní chemie . . . . .	11	<b>SOUT</b>	Předmětové olympiády od A do Z . . . . .	11
<b>YPLS</b>	Filosofie programovacích jazyků . . . . .	12	<b>RAT</b>	Racionalita . . . . .	12
<b>FON</b>	Fonetika . . . . .	12	<b>SS</b>	Shell pro šílence . . . . .	12
<b>HIPPO</b>	Hippo::Web . . . . .	11	<b>DRIGHTS</b>	Svoboda a soukromí v digitálním světě . . . . .	12
<b>IOT</b>	Internet of Things . . . . .		<b>SVG</b>	Vektorizace obrázků . . . . .	12
<b>DRON</b>	Létání a natáčení s dronem . . . . .	11	<b>ZDRAV</b>	Základy první pomoci . . . . .	11
<b>LING</b>	Lingvištika . . . . .	12			