

JARNÍ SOUSTŘEDĚNÍ KSP 2026 – 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 Discord), třeba se najde někdo, kdo by vám o tom rád pověděl. Berte a vychutnávejte!

Přednášky jsou členěné do třech kategorií. Na ty, o kterých si myslíme, že jsou základní a esenciální pro každého programátora, na přednášky pokročilejší, dotýkající se zajímavých oblastí informatiky, a nakonec na přednášky půlnoční, které často rozšiřují obzory až za rámec informatiky. Berte a vychutnávejte!

Údaje o jedné přednášce vypadají asi takto:

Stručný úvod do základů teorie vlkodlaků (*“Za dne ukryt v hloubi lesa, dės 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 (nebo nabídku možných přednášejících, pokud je zde více jmen) a nakonec stručný obsah přednášky. Pokud u přednášky není uveden žádný přednášející, umí ji přednést většina přednášejících a jen vás nechceme unavovat přehledkou našich jmen :-). Hvězdičky znamenají obtížnost.

Základní přednášky

Mezi těmito přednáškami jsou věci, které by měl každý začínající programátor umět. Bez pochopení většiny věcí přednášených na těchto přednáškách se budete na pokročilých přednáškách, které na ně navazují, jen obtížně chytat. Doporučujeme proto nejdříve zvládnout tyto přednášky a osvěžit si nějaký základní programovací jazyk, než se pustíte do pokročilejších věcí. Všechny přednášky ze základní kategorie na soustředění budou, proto nemusíš rozdělit všechny body v této kategorii. Hlasováním nám dej vědět, jak moc už látku z dané přednášky umíš: 0 bodů = všemu rozumím a přednášku nepotřebuji, 20 bodů = vše je pro mě nové.

Základy programování (*“Má $x = x + 1$ řešení?”*)

ZAKL

Úvodní trojdílná přednáška pro ty, kteří mají s programováním jen malé, nebo dokonce žádné zkušenosti. Vysvětlíme si od základů problematiku programování, jako je zápis cyklů, podmínek a funkcí, ukážeme si základní datové typy (n-tice, seznamy, slovníky), datové struktury (fronta, zásobník) a zkusíme si prakticky naprogramovat několik základních algoritmů. Vše se bude ukazovat hlavně na jazyku Python, který je jednoduchý na naučení a přesto zároveň velmi mocný. Jednotlivé přednášky se budou prolínat s přednáškami ZALG.

Základy algoritmizace, složitosti a datových struktur (*“Co by měl každý programátor znát.”*)

ZALG

Kačka Vomelová, Honza Černý

Základní vícedílný kurz algoritmů a datových struktur, který se bude prolínat se ZAKL. Jak poznat, který algoritmus je efektivnější? Přehled základních algoritmů. Co je to datová struktura a několik jejích ukázek. Vše si procvičíme na příkladech.

Grafy & algoritmy (*“Stromy, listy, lesy, pařezy, cesty, kružnice. . .”*)

GA

Kačka Vomelová, Jirka Kalvoda

Spousta algoritmických problémů se dá popsat pomocí teorie grafů. Ukážeme si její základy: co je to graf, jak se dá v programu reprezentovat a k čemu se dá použít. Naučíme se hledat nejkratší cestu v bludišti nebo na mapě.

Základní programovací jazyky a techniky

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

HW

Martin „Medvěd“ Mareš, Jirka Kalvoda

Ukážeme si, proč programy fungují tak, jak jsme zvyklí. Co umí procesor, co dělá paměť a jak se to dá k něčemu použít. Ukážeme si nějaký program v Céčku a v Assembleru a koukneme se, kolik toho řeší Python za nás. Co dělá operační systém, jak je třeba možné, že na jednom procesoru běží najednou několik procesů. Ukážeme si, že počítače jsou překvapivě hloupá stvoření, co umí jenom základní počty, ale na programování nám to stačí.

Jirka Kalvoda

Úvod do Pythonu pro ty, kteří již umí programovat v jiném jazyce. V čem se liší od ostatních jazyků a proč se v něm píše tak snadno. Proč se překládá až při spuštění, jaké výhody a jaké nevýhody to s sebou nese. Letmý úvod do balíčků aneb skoro všechno již někdo napsal za nás.

Pokročilé přednášky

Tyto přednášky by měly jednak dále rozvíjet znalosti ze základních přednášek, ale také nabízet další zajímavé programátorské techniky a technologie, které se mohou každodenně hodit.

Algoritmizace

Datové struktury pro pokročilé * (*“Pojďme na procházku binárním lesem”*)

DS

Martin „Medvěd“ Mareš, Jirka Kalvoda

Přehled šikovných datových struktur, které se nevešly do ZALG. Vyhledávací stromy a různé způsoby jejich vyvažování a „ozdobení“. Hešování aneb hledáme v téměř konstantním čase. Líné datové struktury a amortizovaná složitost.

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

ITREE

Michal Kodad, Jirka Kalvoda, Standa Lukeš

Intervalový strom je datová struktura pracující s intervaly umožňující nám pokládat na nich dotazy třeba na největší prvek v intervalu (a to i když si dovolíme měnit hodnoty). Ukážeme si, jak tato struktura funguje, na co všechno se dá použít, jak ji aktualizovat a jak ji upravovat, aby zvládla ještě víc.

Treap

TREPAP

Michal Kodad, Jirka Kalvoda

Vyvažované stromy jsou úžasná datová struktura, ale většina normálních vyvažovaných stromů je pracné naprogramovat správně. Místo toho si řekneme, že náhoda to za nás vyváží. Sice přijdeme o pěknou worst-case složitost, ale bude se tato datová struktura lehce programovat. Navíc namísto normálních operací na vyhledávacích stromech, tak budeme moct podporovat i intervalové úpravy.

Hledání v textu * (*“Jak najít jehlu v kupce sena a přitom se nebodnout?”*)

TEXT

Vladimír Sklenář, Jirka Kalvoda, Honza Černý

Někdy potřebujeme najít podřetězec ve velkém množství textu. Jak to udělat co nejrychleji? K tomu se nám budou hodit vyhledávací automaty, například Knuthův-Morrisův-Prattův algoritmus. Od toho dojdeme k dalším algoritmům na zpracování textu.

Parsování textu (*“ $1+2*4 = 12$ ”*)

PARSE

Standa Lukeš

Často potřebujeme načíst nějaký složitý textový vstup: matematický výraz, zdroják, konfigurační soubor, . . . Ukážeme si, jak takové textové formáty analyzovat (neboli parsovat), aniž bychom v nich zabloudili: jak si napsat rekurzivní parser od základu i jak použít automatictější nástroje založené na regulárních výrazech a gramatikách.

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

AMORT

Kačka Vomelová, Martin „Medvěd“ Mareš, Jirka Kalvoda

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.

Rekurze (*“Cihla váží kilo a půl cihly . . .”*)

REC

Martin „Medvěd“ Mareš, Jirka Kalvoda

Jedním z nejmocnějších algoritmických nástrojů jsou rekurzivní funkce – tedy takové, které volají samy sebe. Ukážeme, k čemu může být rekurze užitečná a jak se v ní neztratit. A kdy je naopak lepší se rekurzi vyhýbat.

Další programovací jazyky a techniky

Programování v jazyce C

C

Kačka Vomelová, Jirka Kalvoda

Jazyk C je tu s námi již více než 50 let, ale stále patří k velmi používaným jazykům, které se hodí umět. C najde využití od low-level programování přes zpracování videa až po psaní linuxového kernelu, zejména pokud má být něco rychlé. Představíme si datové typy a běžné programové konstrukce, vysvětlíme si základy práce s ukazateli a také se seznámíme se standardními knihovnamy jazyka C.

Základy jazyka C++

cpp

Michal Kodad

Základní exkurze do jazyka C++ - jazyka, po kterém sáhnete, když skutečně potřebujete výkon. Dává totiž daleko větší kontrolu nad tím, co se kdy bude v systému dít a jak. Není to jazyk pro pohodlné programátory, o to více se jeho znalost cení. Vhodné pro kohokoli.

Pokročilý Python (*"I can't comprehend this comprehension."*)

PYTH2

Michal Kodad & David „Dejwut“ Pacák, Jirka Kalvoda, Standa Lukeš

Povídání o tom, co nám Python nabízí, aby se v něm hezky programovalo, které zajde o trošku dále než základní kurz, určené zejména těm, co už v pythonu programovali před soustředěním. Jak psát *pythonicky* aneb *enumerate*, *zip*, *comprehensions* či generátory. Jak se neztratit v balíčcích a jak si vytvořit vlastní. Co za pasti skýtá Python a jak přesně fungují pythonní třídy. Jak se dívat na pythonní objekty z pythonu a případně jak za běhu programu modifikovat asi tak cokoliv a proč to možná nechceme dělat.

Předpoklady: Základy Pythonu

Asynchronní programování (*"Kdo čeká, ten se dočká. A tím spíš, pokud čeká víckrát naráz."*)

ASYNC

Jirka Kalvoda, Standa Lukeš

Jistě jste se setkali s programy, co například čekali na vstup od uživatele. Co když ale chceme čekat na více věcí současně? Například krom odpovědi od uživatele může dorazit něco po síti, z jiného programu nebo ze souboru. Na této přednášce si ukážeme různé způsoby, jak tyto problémy řešit. Od aktivní smyčky přes low-lever rozhraní linuxího kernelu – rodiny *poll*, přes programování za pomoci *callbacků* po asynchronní programování v Pythonu (pomocí knihovny *trio*).

Paralelní výpočty v MPI a OpenMP

PAR

Lukáš Veškrna

Vysvětlíme si, jaký je rozdíl mezi *shared* a *distributed memory* paralelními modely. Následně se podíváme na dva frameworky pro paralelní výpočty: *OpenMP* (*shared memory*) a *MPI* (*distributed memory*). Budeme hledat zákeřné chyby v programech v jazyce *C* s *OpenMP* a *MPI* a opravovat je.

Programování na grafické kartě * (*"Řídí se to jako raketa – létá rychle, ale nemá volant."*)

GPU

Kuba Pelc

Dnes již není grafická karta jen placka převádějící digitální pixely na analogový signál. Dá se na ní počítat kde co. Zde si představíme trochu technologie *CUDA* a *compute shadery* v *OpenGL* 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*.

TypeScript (*"Typový heavy metal!!!"*)

TS

Kuba Pelc, Marian Šámal

TypeScript je nadstavba nad *JavaScriptem*, která do něj přidává statické typování. A nejen ledažaké! Umí spoustu pěkných věcí, jenž v ostatních mainstream jazycích nenajdeme. Ukážeme si, jak v něm pohodlně popsat i velmi složité koncepty, a jak moc nám tento jazyk zpříjemní psaní webových věcí oproti *JavaScriptu*, který je notorický svou náchylností na nenápadné chyby, které odhalíte, až jakmile vám aplikace spadne.

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

OOP

Michal Kodad, Honza Černý

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*.

Jak funguje web? (*"Tři jazyky, nekonečno možností"*)

WEB

Jakub Hampl, Jirka Kalvoda, Marian Šámal

Web je čím dál tím mocnější a rozšířenější platforma. Dnes existuje nespočet způsobů, jak si vytvořit aplikaci žijící na internetu, my se spolu podíváme na stavební kameny: *HTML*, *CSS* a *JavaScript*. Zjistíme, co je to *DOM*, proč si nejsou různé prohlížeče rovny a co jsou to ty zatracené sušenky, co po nás všichni chtějí. *WASM*, *CORS*, *PWA* a další zkratky. Co vše zajímavého lze vykuknout z *developer tools*. Nahlédneme do křišťálové koule a pokusíme se předpovědět, jak to s webem bude dál.

Vyrábění webových stránek v Jekyllu. (*"Nebud' internetový bezzemek!"*)

JEKYLL

Jakub Hampl, Honza Černý

Na této praktické přednášce si pomocí *Jekyllu* společně vytvoříme jednoduchý statický web. Bude se hodit znát základy *HTML* a *CSS* (ale není to nutná podmínka). Naučíme se používat *templaty* a navrhovat stránky tak, aby byly hezké jak na počítači, tak i mobilním zařízení.

Předpoklady: Základní povědomí o HTML a CSS.

Logické programování (*"Mohu být svým vlastním dědečkem?"*)

LOGP

Honza Černý

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 neobyčejně zjednoduší a které naopak programování promění v noční můru. Pokud jsi milovník *rekurze*, budeš u této přednášky nejspíš skoro celou dobu spokojeně vrnět.

Umělá inteligence

Herní algoritmy * (*“Když nemáte na to, abyste vyhráli šachový turnaj. . .”*)

AIGAME

Michal Kodad

Povídání o tom, jak programovat počítačové soupeře do šachů a her jim podobným. Základní minimaxový algoritmus a jeho vylepšení neboli α - β ořezávání. Stále pomalé? Několik nápadů na efektivnější ořezávání. Ne u všech her však funguje hrubá síla (minimax) dobře, ukážeme tedy ještě pravděpodobnostní přístup Monte Carlo Tree Search.

Neuronové sítě * (*“Jak si postavit vlastní neuronovou síť a naučit ji číst”*)

NN

Vladimír Sklenář, Michal Kodad

Umělá inteligence je dneska zastřešující termín pro až děsivé množství věcí. Někde v kořenech tohoto spleteného lesa jsou neuronové sítě, na kterých stojí mocné stromy velkých modelů jako třeba všeznáma Chat-GPT. Naneštěstí většina AI tutoriálů vám dá všemocné nástroje, aniž by vám ukázaly co se v nich skutečně děje. Pojďme tedy společně postavit neuronku jen za pomoci trocha násobení a sčítání! A možná ji naučíme i něco dělat a ne jen smutně existovat. . .

Transformery **

NEURO2

Michal Kodad

Tato navazující přednáška na Neuronové sítě se zabývá transformery, což je pokročilý model pro zpracování přirozeného jazyka. Transformery se staly revolučním nástrojem v oblasti strojového učení a jsou základem mnoha moderních aplikací, jako jsou strojový překlad, rozpoznávání řeči nebo generování textu. Během přednášky se seznámíte se základními principy transformerů, jako je self-attention mechanismus a enkodér-dekodér architektura. Budeme také diskutovat o jejich výhodách a omezeních a představíme některé příklady jejich úspěšného využití v různých oblastech. – Anotaci napsal ChatGPT

Přírodou inspirované algoritmy * (*“Kolik mravenců potřebujeme aby spočítali NP těžký problém ?”*)

NATURALG

Honza Černý

Příroda je nádherná a celá tisíciletí se jí inspirujeme. Kolik už inspirovala spisovatelů, básníků a malířů. Nyní jsou na řadě programátoři. Když si nebudeme vědět s nějakým těžkým problémem, tak zkusíme nenápadně opisovat od přírody.

Umělá inteligence * (*“Jak napsat program, co vás porazí v šachách.”*)

AI

Honza Černý

Ukážeme si, jak počítače přemýšlí při řešení problémů a jakým způsobem hledají řešení. Volně se dostaneme k prohledávání stavového prostoru (který bývá exponenciálně velký) a ukážeme si různé jak informované, tak neinformované techniky pro jeho procházení. Setkáme se třeba s algoritmy, které jsou použity v GPS.

Bezpečnost umělé inteligence * (*“Jak zabránit počítačům, aby k nám poslali z budoucnosti nahého Arnolda Schwarzeneggera?”*)

AISA

Honza Černý

Umělá inteligence se k nám blíží každým dnem. S ní se ale blíží i spousta reálných rizik. Podívejme se společně na reálná nebezpečí, která s umělou inteligencí přicházejí a jestli jim dokážeme zabránit.

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

EVA

Honza Černý

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 pohyblivých se ve velkých prostorech řešení.

Počítačová lingvistika

Měřit čitelnost textů a jak na tom je česká byrokracie

PONK

Honza Černý

Projekt PONK je projekt vedený na ÚFALu, který se zabývá vývojem nástrojů pro analýzu čitelnosti textů. Tyto nástroje mají pomoci najít nečitelné texty v české byrokracii a zlepšit jejich čitelnost. Na přednášce si ukážeme metody, které byly pro PONK vyvinuty, a vysvětlíme, jak fungují. Dozvíte se také jak vypadá vývoj lingvistických nástrojů a kde si je můžete vyzkoušet.

Počítačová lingvistika * (*“Co dobrého může vzejít z propojení informatiky a lingvistiky”*)

NLP

Honza Černý

Pojďme společně nahlédnout pod pokličku počítačové lingvistiky a počítačovému zpracování přirozeného jazyka. Podíváme se na vlastnosti přirozených jazyků a povíme si, jak moc jednotlivé vlastnosti komplikují počítačové zpracování. Ukážeme si také několik praktických aplikací počítačové lingvistiky – kontrolu pravopisu, strojový překlad a vyhledávání v dokumentech podle témat atd.

Grafické programování a rendering

Shadery * (*“Malujeme kódem”*)

SHADER

Kuba Pelc

Programování nemusí být jen o terminálech a webech. Ukážeme si, jak i z několika málo řádků kódu může vzniknout něco hezkého a barevného. Shadery jsou krátké programky, které každému pixelu obrázku přiřadí nějakou barvu. Podíváme se s jejich pomocí do hlubin fraktálů a z trošky náhody uděláme mraky nebo vodní hladinu. Přednáška bude převážně praktická.

Kuba Pelc

Vyhlídkový výlet světem pomalé fotorealistické počítačové grafiky (na rozdíl od rychlé herní grafiky v GAMEGFX), demonstrováný na :hippo_god: (viz <https://www.shadertoy.com/view/wtGczK>). Projdeme základní principy simulace světla, raytracingu, GPU a proceduální generace. Jak definovat scénu s hrochem pomocí pár řádek kódu a jak to celé použít pro nakreslení obrázku v Shadertoy.

Grafika v počítačových hrách (“*But can it run Crysis?*”)**GAMEGFX***Kuba Pelc*

Vyhlídkový výlet světem rychlé herní počítačové grafiky (na rozdíl od pomalé fotorealistické v HIPPOGOD). Dozvíte se, jak se vyráběly hezké pixely rychle a efektivně na různých érách hardwaru i softwaru, od konce devadesátek až po současnost. Jak funguje vykreslování a GPU zevnitř a co jsou všechny ty obskurní zkratky v grafických nastaveních. A taky pár her rozpitváváme zaživa frame debuggerem.

Počítače, sítě, systémy

Základy sítí (“*Jak si postavit Internet*”)**NET***David „Dejwut“ Pacák, Martin „Medvěd“ Mareš, Jirka Kalvoda*

Jak fungují počítačové sítě: od elektronů v drátech (nebo fotonů v optických kabelech) přes komunikaci na jedné malé síti až ke komunikaci v celém Internetu. Vysvětlíme si rámce, pakety, MAC a IP adresy, routování v malých i ve velkých sítích. Jak to reálně funguje s IPv4 a NATem, co to jsou porty a jak se od sebe liší TCP a UDP. A jak postupně zavádíme IPv6.

Sítě II – protokoly (“*Jak si přečíst mailu. . . sousedovy mailu.*”)**NET2***David „Dejwut“ Pacák, Martin „Medvěd“ Mareš, Jirka Kalvoda*

Volné navázání na NET aneb máme fungující síť a chceme nad ní provozovat složitější komunikaci. ICMP aneb servisní protokol Internetu, DNS a překlad doménových jmen, jednoduché textové protokoly jako je FTP, SMTP, IMAP nebo nejpoužívanější webové HTTP. U HTTP se zastavíme trochu déle – hlavičky, návratové kódy, cookie, více domén na stejné IP adrese a SSL certifikáty.

*Předpoklady: Základní povědomí o počítačových sítích v rozsahu NET***Jak poslepoval internet** (“*Když se perfektní protokoly střetnou s neperfektní realitou.*”)**INT***Jakub Hampel*

Kde se vlastně berou IP adresy? Kde a jak vznikají top-level domény? Co vlastně dělají poskytovatelé internetu? Na této přednášce si povíme něco o protokolu BGP, autonomních systémech a organizacích IANA a RIPE NCC. Podíváme se na globální infrastrukturu internetu trochu z dálky a dojde i na špetku historie.

*Předpoklady: Povědomí o počítačových sítích v rozsahu NET***Základy Linuxu** (“*Kolik příkazů umíš, tolikrát jsi tučňákem!*”)**LINUX***Marian Šámal, Jirka Kalvoda*

Každý programátor někdy bude muset používat příkazový řádek, často to nejde jinak a často je to nejpohodlnější. Vyzkoušíme si základní příkazy na manipulaci souborů a editaci textu, nainstalujeme si všechn software na světě, upravíme si to k obrazu svému a budeme to dělat stovky kilometrů daleko a možná i na Windowsu.

Linux pro správce serveru (“*Printer is on fire???*”)**SERVER***Vladimír Sklenář, Martin „Medvěd“ Mareš, Jirka Kalvoda*

Jak vytvořit jednoduchý Linuxový server, který poskytuje služby vaší domácnosti, nebo třeba nějaké větší síti. Co se tam hodí provozovat? Povíme o SSH, klíčích, šifrování, systemd, Apache a Nginxu, nastavení mailového serveru i DNS. Jak server zabezpečit před útočníky, jak před ztrátou dat a jak před uklízečkou. Vše si vyzkoušíme prakticky, třeba na virtuálním počítači.

*Předpoklady: Základní znalost Linuxu.***Mikrokontroléry** (“*Nejlepší debugger je LEDka.*”)**MCU***Martin „Medvěd“ Mareš, Jirka Kalvoda*

Srdcem mnoha dnešních technických hraček je mikrokontrolér. To je čip, na kterém je integrovaný nejen procesor, ale i paměť a spousta zajímavých periférií. Ukážeme si, jak se mikrokontroléry programují, jaké periferie typicky obsahují a jak je používat ke komunikaci s okolním světem. Něco si vyzkoušíme i prakticky na STM32.

Předpoklady: Hodí se základní znalost jazyka C.

Vývoj software

Systém pro správu verzí Git (“*U svatýho tučňáka, kdo sem napsal tohle? Ono to tvrdí, že JÁ?!*”)**GIT***Marian Šámal, Michal Kodad, Jirka Kalvoda*

Když se něco vyvíjí delší dobu a/nebo v týmu, přijde vhod nějaký nástroj, který umí ukládat různé verze zdrojového kódu, zjistit kdo co přidal a proč, k nějaké verzi se vrátit nebo třeba vrátit jenom jednu změnu, kterou udělal kamarád před rokem. Git je dnes nejpoužívanější takový nástroj. Povíme si, jak ukládá změny, co jsou commity, větve, tagy a jak vypadá merge mezi větvemi. Nakonec zkusíme předvést i nějaké užitečné triky: třeba hledání bugů půlením historie.

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

VIM

Marian Šámal, Martin „Medvěd“ Mareš, Jirka Kalvoda

Odložme na chvíli své myšle 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.

Aplikace informatiky a matematiky

Kryptografie (“Gbgħ arav zbp gnwan mcenin.”)

CRYPT

Martin „Medvěd“ Mareš, Jirka Kalvoda

Kryptografie se zabývá šiframi, jejich konstrukcí a zejména jejich luštěním. Začneme se symetrickými a asymetrickými šiframi a jednosměrnými funkcemi. Z nich pak vybudujeme složitější kryptografické protokoly na bezpečný přenos, autentikaci a digitální podpisy. Vymyslíme dokonce, jak si hodit korunou po telefonu, a také předvedeme nerozluštitelnou šifru a dokonce to o ní dokážeme.

Unicode (“Jaký kód má sněhulák s kudrnatými vlasy?”)

UNI

Martin „Medvěd“ Mareš, Standa Lukeš

Jak funguje znaková sada Unicode, která se snaží zapsat všechny jazyky světa? Codepointy versus glyfy. Kombinující znaky, čtvero normálních forem a pátá lehce nenormální. Typografické a neviditelné znaky. Co všechno prozradí Unicode Character Database. Uložení v paměti: formáty UCS-2, UCS-4, UTF-8 a UTF-16, nešvar s BOM. Tajemný svět emoji. Jak se s Unicode programuje? A jako vždy: bezpečnostní problémy.

Zpracování dat (“Bez práce nejsou koláč...ové grafy.”)

DATA

Vladimír Sklenář, Standa Lukeš

O světě jde sehnat spousta zajímavých dat ve strojově zpracovatelné podobě: obce a domy v nich, linky hromadné dopravy, katalogy hvězd, slova v češtině, katalog pokémonů, ... Pojďme se podívat, jak s daty zacházet. Naučíme se číst různé formáty dat od CSV až po XML, data zkoumat, filtrovat a kreslit podle nich pěkné grafy. Vyzkoušíme si to prakticky v Pythonu na reálných datech, které si umíte veřejně stáhnout.

Předpoklady: PYTH

Buněčné automaty a Game of Life (“Čtverečkováný svět, co není Minecraft”)

LIFE

Martin „Medvěd“ Mareš

Game of Life je dvojrozměrný svět, ve kterém se buňky vyvíjí podle průzračně jednoduchých pravidel. Už desítky let v tomto světě objevujeme další a další zajímavé jevy. Tak do něj také nahlédneme, prozkoumáme souvislosti s evoluční biologii i s algoritmy. Též uvidíme, jak Život zapadá do obecnějšího světa buněčných automatů.

Webové mapy (“Země je perfektní jednotkový čtverec, že ano?”)

WEBMAP

Kuba Pelc

Webové mapy - takřka všudypřítomné a velmi užitečné. Jak ale fungují? No a co teprve, když chceme mapy vektorové? A co když chceme zobrazovat Zemi jako kouli? A zároveň vektorové? Zdánlivě jednoduchá věc jako mapa totiž dokáže skrývat mnoho zajímavých technických detailů. Nahlédneme pod pokličku knihoven Leaflet a především MapLibre, které nedávno získalo podporu vektorového glóbu. Možná jsem v tom měl prsty...

Komprese dat

ZIP

Standa Lukeš

Přehled základních kompresních algoritmů: triviální algoritmy (RLE), statistické metody (Huffmanovo a aritmetické kódování), slovníková komprese (LZ77, LZ78, LZW). Různé kódování dat, aby standardní kompresní algoritmy byly šťastné (prediktory, delta, bit split).

Matematické přednášky

Lineární algebra pro začátečníky

LIN1

Ondra Chwiedziuk, Lukáš Veškrna, Honza Černý

Lineární algebra je nejdůležitější partie matematiky, jež nachází využití napříč obory, jmenovitě ve strojovém učení, vyhledávání (PageRank) či v kryptografii. V této přednášce se zaměříme na vybudování základů maticového počtu, ale i trochu abstraktnější pohled na problematiku.

Lineární algebra pro mírně pokročilé

LIN2

Ondra Chwiedziuk, Lukáš Veškrna

Lineární algebra je nejdůležitější partie matematiky, jež nachází využití napříč obory, jmenovitě ve strojovém učení, vyhledávání (PageRank) či v kryptografii. V pokračování úvodní přednášky do lineární algebry se zaměříme na diagonalizaci a ortogonalizaci.

Předpoklady: LIN1

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

KOMB

Martin „Medvěd“ Mareš, Jirka Kalvoda

Pokusíme se vybudovat kombinatoriku intuitivně. Co nejvíce se vyhneme počítání se vzorci, vystačíme si s elegantními úvahami. Kromě základních technik si ukážeme, jak nám mohou pomoci rekurence a jak se úlohy dají převádět mezi sebou. Procvičíme na spoustě příkladů.

Ondra Chwiedziuk, Jirka Kalvoda

Pohled na teorii grafů trochu teoretičtěji. Začátek budeme věnovat základům, načež se přesuneme na kreslení grafů na plochy a jejich barevnost. Hlavním výsledkem bude obarvení rovinných grafů 5 barvami. Také si něco řekneme o tom, jak poznat, zda jsou 2 grafy stejné (izomorfní).

Lineární programování

LINPRG

Ondra Chwiedziuk, Jirka Kalvoda

Jsme na počátku 40. let v New Yorku na Manhattanu a snažíme se vést pekařství. Víme, že každý den se prodá vše, co upěčeme, avšak každý den zvládneme obstarat jen omezené množství mouky, vajec a mléka. Jaké druhy pečiva máme upéct, abychom měli co největší profit? To a na spoustu dalších otázek si zodpovíme v této přednášce. Ukážeme si, jak problém popsat, vysvětlíme si metodu řešení a jak to správně strčit do počítače, aby to spočítal za nás.

*Předpoklady: LIN1***Teorie her**

GT

Ondra Chwiedziuk

Určitě jste hráli následující "hru": jedete autobusem, zastávka, kde vystupujete, je však na znamení. Vám se však nechce zmáčknout tlačítko, protože je na druhé straně autobusu. Tak čekáte, jestli to zmáčkne někdo jiný. Jenže čím blíže jste zastávce, tím víc je jasné, že ani ostatním se tlačítko nechce zmáčknout. Zvednete se, nebo necháte projet autobus bez zastavení? V přednášce si ukážeme, jak tuto a další podobné "hry" lze matematicky popsat a následně určit vítěznou strategii. Aplikace těchto přístupů jsou od volebních systémů po routování mezi sítěmi.

Těžké problémy

HARD

Ondra Chwiedziuk, Martin „Medvěd“ Mareš, Jirka Kalvoda

V rámci této přednášky se budeme zabývat problémy tak těžkými, že nikdo na světě pro ně neumí vymyslet efektivní (rozuměj polynomiální) algoritmus. Spousta lidí dokonce věří, že to vůbec možné není. Abychom mezi tyto problémy pronikli, seznámíme se s pojmy NP-úplnosti a NP-těžkosti. Především si však konkrétní těžké úlohy ukážeme a naučíme se i některé těžké úlohy rozpoznat. Závěrem si řekneme, jak se s těžkými úlohami vypořádat v praxi.

Teorie čísel a RSA (*“Pošli mi zprávu, tady máš klíč.”*)

RSA

Ondra Chwiedziuk, Martin „Medvěd“ Mareš

Řekneme si, čím se zabývá teorie čísel a vysvětlíme si její nejznámější aplikaci: šifru RSA. Vybudujeme teorii kolem dělitelnosti, řekneme si základní větu aritmetiky, vysvětlíme si Eukleidův algoritmus, Bézoutovu rovnost, Eulerovu funkci a důležité vlastnosti prvočísel. Když budeme znát všechny tyto pojmy, můžeme se pustit do vysvětlení, proč vlastně RSA funguje. Na závěr si řekneme slabiny RSA a proč už se dneska přestává používat.

Diferenciální rovnice (*“Jak si napsat simulaci?”*)

DIFF

Lukáš Veškrna

Zajímá vás jazyk, kterým hovoří fyzika? A chtěli byste umět simulovat fyzikální úlohy na počítači? Probereme základy derivací a integrálů, pak si ukážeme nějaké diferenciální rovnice a jejich roli ve světě (biologie, fyzika, chemie, finance). Nakonec se podíváme na základy numerického řešení diferenciálních rovnic na počítači s několika ukázkami.

Ostatní přednášky

Debugování v IDE a bez něj (*“Proč tam ty chyby dáváte když je pak musíte hledat?”*)

DEBUG

Vladimír Sklenář, Standa Lukeš

Určitě se vám někdy stalo, že jste měli v programu chybu, ale nevěděli jste, kde přesně je. Na této přednášce si ukážeme, jak používat různé nástroje pro debugování, jako jsou breakpoints, watchpointy, stack trace a další. Také se podíváme na to, jak debugovat programy bez použití IDE, například pomocí gdb nebo stderr výstupu. Nakonec si zkusíme zdebugovat několik jednoduchých programů v C++ a Pythonu.

Pitfalls programování (*“Jak otevřít dveře a nepustit dovnitř vlka nebo rybu?”*)

PITFAL

Vladimír Sklenář, David „Dejwut“ Pacák

Jak se vyhnout chybám, které jsou na první pohled neviditelné, ale mohou způsobit velké problémy? Zaměříme se na nebezpečí přetečení bufferu, SQL injection nebo XSS útoky. Společně se zamyslíme nad tím, co vlastně znamená bezpečný software a jak ho navrhnout. Nakonec si ukážeme několik konkrétních příkladů a vyzkoušíme si i vlastní útok na jednoduchou aplikaci. Přednáška bude spíše diskusní a praktická.

Soukromí a bezpečnost na webu (*“A kdy teda dorazí ty sušenky?”*)

WEBPRIV

Standa Lukeš

Jak funguje web z pohledu soukromí uživatelů, a co dělat pro zlepšení své situace. Použití webové aplikace je dnes sice typicky nejlepší možnost jak používat nepříliš důvěryhodný software, ale je dobré vědět co na vás existuje za nástrahy. A jak se proti nim efektivně, a rozumně pohodlně, bránit. Není moc soukromí bez bezpečnosti, takže i jak se nenechat hacknout bude téma.

Programátoři versus realita (*“Jeniffer Null ze Scunthorpe by mohla vyprávět . . .”*)

FALSE

Martin „Medvěd“ Mareš

Sbírka představ o světě, kterým většina programátorů věří, ale vlastně nejsou pravdivé. Jsou všechny dny stejně dlouhé? A všechny minuty? Kolik znaků má jeden znak? Čím jsou oddělené comma-separated values? Kde ve Zlíně je tř. Tomáše Bati č.p. 202? Kolikrát že? Má každý občan EU příjmení? A jméno? A právě jedno rodné číslo? Kolik zavináčů může obsahovat e-mailová adresa?

Půlnoční přednášky

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

Rubikova kostka bez algoritmů

RUBIK

Ondra Chwiedziuk

Už jste někdy zkoušeli složit Rubikovu kostku? A nešlo vám to? Našli jste si návod na internetu, ale byla to jen náhodná směť písmenek? Tak si ukážeme, jak ji složit bez toho. A také si ukážeme, jak Rubikova kostka souvisí s teorií grup.

Pilotování dopravních letadel

LAND

David „Dejwut“ Pacák

Zajímalo vás někdy, co se děje v kokpitech dopravních letadel? Co před sebou piloti vidí, čím letadla ovládají, jak vlastně dokážou přistát v noci nebo za husté mlhy? Popovídáme si o tom, jak se pilotují moderní dopravní letadla, co mají za zajímavé systémy, jak s nimi piloti interagují a o cokoli dalšího bude zájem (a budu o tom dost vědět). Podobná přednáška byla na Smršti, ale nebojte se případného překryvu, mnoho se toho nestihlo.

Demoscéna - galerie programátorského umění (*“I’ve seen things you people wouldn’t believe.”*)

DEMOSCENE

Kuba Pelc

Když se počítače poprvé staly dostupné pro běžné smrtelníky, nic moc ještě neuměly. Ale měly k sobě přibalený manuál od nějakého programovacího jazyka. A tak lidi začali programovat. A začali si *hrát*. Psali malé hry, ale také programky, které kreslily něco hezkého jen tak. Takzvané „intra“ nebo „dema“. A pak se začali předhánět a soutěžit v tom, kdo udělá lepší, hezčí, technicky propracovanější demo. Někdy s limitem na velikost 64 kb, někdy 4 kb, někdy bez limitu. A tak se v průběhu desetiletí vyvinula demoscéna. Pojďme se podívat na ty nejhustější dema! A možná si i říct něco o tom jak fungují.

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

LING

Martin „Medvěd“ Mareš

Převážně nevážné a mírně nepřed-vídatelné po-vídaní 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 třeba mlasknutí &c.

Quo vadis, češtino? (*“Muž žena skopce dokopce si píská.”*)

CZLANG

Martin „Medvěd“ Mareš

O češtině časů minulých, současných i (potenciálně) budoucích. Kde se čeština vzala a jak souvisí s ostatními evropskými jazyky. Co jsme během staletí získali, o co jsme přišli a o co nejspíš brzy přijdeme. A je to škoda? Vykrádáme cizí jazyky a ony zase nás. Kolik je češtin a která je ta správná? Proč už Jan Hus spílal Pražanům? Pojďme navrhnout pravopisnou reformu!

Teorie množin (*“Jablka a hrušky se dají nejen sčítat, ale třeba i násobit.”*)

TEMNO

Martin „Medvěd“ Mareš

Základoškolský přístup „množina je kupříkladu miska jablíček“ nabízí spoustu otázek: Když jablíčka přesuneme do sáčku, bude to stále tatáž množina? A co když kousek jablíčka ukousneme? V rámci této přednášky se pokusíme o vybudování teorie množin od základů (rozuměj axiomů) a to v duchu Zermelo-Fraenkelovském. Pak uvidíme, jak na teorii množin vystavět zbytek matematiky.

Čaj (*“Jak vypadá odvar z nezralých pražců?”*)

TEA

Martin „Medvěd“ Mareš, Jirka Kalvoda

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.

FPV droník (*“O dimenzi lepší než jen řídit auto.”*)

DRON

Jakub Hampl

Malé drony, takzvané *Tinywhoops*, jsou kvadrokoptéry o velikosti dlaně, se kterými je kopec srandy. Ukážeme si, jak s dronem létat, jak ho nastavit, opravit a jak se o něj a o příslušenství starat, aby dlouho vydrželo. Řekneme si něco málo o protokolu ELRS a analogovém videu. Budete si moci vyzkoušet simulátor a pokud budou podmínky příznivé, zalítáme si i venku.

Honza and Kuba yell at cloud (*“Dnešní svět = 1984 ∪ Fahrenheit451 ∪ Kocourkov”*)

CLOUDS

Jakub Hampl, Honza Černý

Podobně jako v Božské komedii projdete několika kruhy pekla představující dnešní svět technologií. Přestože obsahuje mnoho různých doporučení, byla vytvořena převážně k tomu, aby se nad danou problematikou člověk sám zamyslel.

Nová náboženská hnutí (*“Vše, co potřebujete vědět, abyste si mohli založit sektu.”*)

CULT

Honza Černý

Od poloviny minulého století přibýlo mnoho nových náboženství a duchovních hnutí. Povíme si o jejich historii a dopadu na společnost. Povíme si i o psychologii schovanou za sektami a jak vlastně interně fungují. Povíme si i o charismatizaci a jak ji poznat.

Hnutí nového věku (*“Jak si vyrobit boha na míru.”*)

NEWAGE

Honza Černý

Žijeme v době, kdy můžeme být svědkem velkých změn, a to dokonce i ve vnímání duchovna a životních hodnot. Řekneme si o historickém vývoji náboženského směru „new age“ a jeho aktuální podobě. Cílem přednášky je pochopit vývoj hodnotových systémů a jak ovlivňuje jednotlivce. Přednáška z velké části vychází z pozorování Karla Gustava Junga.

Hlasitá přednáška RELOADED (*“Jak nehoda při obsluze soustruhu založila hudební žánr.”*)

LOUD

Honza Černý

Tato přednáška je chronologickým a snad i trochu systematickým náhledem na vývoj metalové hudby a žánrů z nich odvozených. Přednáška bude nejen o hudbě, ale i příbězích o jejich interpretech. Přednáška nejen pro posluchače metalové hudby, ale i pro lidi, kteří se s ní nikdy nesetkali a chtěli by nahlédnout do subkultury okolo zkreslených tónů kytar.

Abecední seznam přednášek

Základní přednášky

GA	Grafy & algoritmy	1	ZALG	Základy algoritmizace, složitosti a datových struktur	1
HW	Principy počítačů	1	ZAKL	Základy programování	1
PYTH	Python	2			

Pokročilé přednášky

AMORT	Amortizace	2	PAR	Paralelní výpočty v MPI a OpenMP	3
ASYN	Asynchronní programování	3	PARSE	Parsování textu	2
AISAFETY	Bezpečnost umělé inteligence	4	PITFAL	Pitfalls programování	7
LIFE	Buněčné automaty a Game of Life	6	NLP	Počítačová lingvistika	4
DS	Datové struktury pro pokročilé	2	PYTH2	Pokročilý Python	3
DEBUG	Debugování v IDE a bez něj	7	NATURALGP	Přírodou inspirované algoritmy	4
DIFF	Diferenciální rovnice	7	FALSE	Programátoři versus realita	8
EVA	Evoluční algoritmy	4	GPU	Programování na grafické kartě	3
HIPPOGOD	Fotorealistická grafika aplikovaná na hrochy	5	C	Programování v jazyce C	2
GAMEGFX	Grafika v počítačových hrách	5	REC	Rekurze	2
GRAFY	Grafy bez algoritmů	7	SHADER	Shadery	4
AIGAME	Herní algoritmy	4	NET2	Sítě II – protokoly	5
TEXT	Hledání v textu	2	WEBPRIV	Soukromí a bezpečnost na webu	7
ITREE	Intervalové stromy	2	GIT	Systém pro správu verzí Git	5
WEB	Jak funguje web?	3	RSA	Teorie čísel a RSA	7
INT	Jak posleovat internet	5	GT	Teorie her	7
KOMB	Kombinatorika	6	VIM	Textový editor Vim	6
ZIP	Komprese dat	6	HARD	Těžké problémy	7
CRYPT	Kryptografie	6	NEURO2	Transformery	4
LIN2	Lineární algebra pro mírně pokročilé	6	TREAP	Treap	2
LIN1	Lineární algebra pro začátečníky	6	TS	TypeScript	3
LINPRG	Lineární programování	7	AI	Umělá inteligence	4
SERVER	Linux pro správce serveru	5	UNI	Unicode	6
LOGP	Logické programování	3	JEKYLL	Vyrábění webových stránek v Jekyllu.	3
PONK	Měřit čitelnost textů a jak na tom je česká byrokracie	4	WEBMAP	Webové mapy	6
MCU	Mikrokontroléry	5	cpp	Základy jazyka C++	2
NN	Neuronové sítě	4	LINUX	Základy Linuxu	5
OOP	Objektově orientované programování	3	NET	Základy sítí	5
			DATA	Zpracování dat	6

Půlnoční přednášky

TEA	Čaj	8	LING	Lingvistika	8
DEMOSCENE	Demoscéna - galerie programátorského umění	8	CULT	Nová náboženská hnutí	9
DRON	FPV droník	8	LAND	Pilotování dopravních letadel	8
LOUD	Hlasitá přednáška RELOADED	9	CZLANG	Quo vadis, češtino?	8
NEWAGE	Hnutí nového věku	9	RUBIK	Rubikova kostka bez algoritmů	8
CLOUDS	Honza and Kuba yell at cloud	8	TEMNO	Teorie množin	8