

Procedurální programování

Jan Faigl

Katedra počítačů

Fakulta elektrotechnická

České vysoké učení technické v Praze

Organizace předmětu

B0B36PRP – Procedurální programování

Přehled témat

- Část 1 – Organizace předmětu
 - Organizace
 - Prostředky dosažení cílů
 - Hodnocení a zkouška
 - Komunikace
 - Vývojová prostředí a služby akademické sítě

Část I

Organizace předmětu

Předmět a přednášející

B0B36PRP – Procedurální programování

- Webové stránky předmětu <https://cw.fel.cvut.cz/wiki/courses/b0b36prp>
- Odevzdávání domácích úkolů <https://cw.felk.cvut.cz/brute>
- Přednášející:

- prof. Ing. **Jan Faigl**, Ph.D.



- Katedra počítačů – <https://cs.fel.cvut.cz>
- Centrum umělé inteligence – **Artificial Intelligence Center (AIC)**

<https://aic.fel.cvut.cz>

- Centrum robotiky a autonomních systémů
Center for Robotics and Autonomous Systems – CRAS

<https://robotics.fel.cvut.cz>

- **Laboratoř výpočetní robotiky (Computational Robotics Laboratory)**

<https://comrob.fel.cvut.cz>

Cvičící

- Ing. **Rudolf Jakub Szadkowski**, Ph.D.



- RNDr. **Ingrid Nagyová**, Ph.D.



- Ing. **Jindřiška Deckerová**



- Ing. **Jáchym Herynek**



- prof. Ing. **Jan Faigl**, Ph.D.



Organizace a hodnocení B0B36PRP – Procedurální programování

- Rozsah: 2p+2c; Zakončení: Z,ZK; Kredity: 6; 1 ECTS kredit je 25–30 hodin za semestr, cca 180 h.

- Kontaktní část (přednáška a cvičení): **3 hodiny týdně**, tj. 42 hodin celkem.

- Zkouška včetně přípravy: **10 hodin**.

- Domácí příprava (úkoly) cca **9 hodin týdně**.

Medián zátěže!

- **Průběžná práce v semestru – Formativní výuka** - domácí úkoly, kvízy, test (zpětná vazba).

- **Zkouška (test a implementace) - Certifikační typ zkoušky** - uspěl(a)/neuspěl(a).

Schopnost samostatné práce na počítačích v učebnách.

- Docházka na **cvičení** a odevzdání domácích úloh.

Samostatná práce (kontrola plagiátů).

- Postupujte systematicky, budete tak postupně rozvíjet své schopnosti.

- Využijte čas v prvních úlohách a naučte se psát programy správně.

Program musí být nejen správný a funkční, ale také čitelný a udržitelný!

- **Konzultace – Je normální nevědět, proto se učíte**, ale netrapte se dlouho sami **konzultujete** s cvičícím/přednášejícím.

Čtete (učebnici), pochopte principy (nejen hledat řešení), hlídejte si čas a včas konzultujte!

- **Maximálně využijte kontaktní čas na cvičení/přednášce, ptejte se, diskutujte!**

- „**Alternativní**“ absolvování předmětu pro velmi zkušené – **Seminář ACM** - předmět A4B36ACM!

Program je „recept“

- Program je „recept“ – posloupnost kroků (výpočtů) popisující průběh řešení problému.
- Programování je schopnost **samostatně**
 - **tvorit programy**;
 - **dekomponovat** úlohy na menší celky;
 - sestavovat z **dílkách částí větší programy** řešící komplexní úlohu.

B0B36PRP – je příležitostí, jak se těmto schopnostem naučit!

Co vám pomůže!

■ Knihy (učebnice)

Základní učební text „Programming in C“ (Kochan, 2014)



Programming in C, 4th Edition,
Stephen G. Kochan, Addison-Wesley, 2014,
ISBN 978-0321776419



C Programming: A Modern Approach, 2nd Edition, *K. N. King*,
W. W. Norton & Company, 2008, ISBN 860-1406428577



The C Programming Language, 2nd Edition (ANSI C), *Brian W.
Kernighan, Dennis M. Ritchie*, Prentice Hall, 1988

1st edition 1978



- Přednášky – podpora učebního textu, slidy, videa, poznámky a **vlastní poznámky**.

Součástí přednášek jsou také zdrojové kódy s příklady!

- Cvičení – získání praktických dovedností řešením domácích úkolů a dalších úloh.

programovat, programovat, programovat

Není jeden styl (předávání znalostí)



Practical C Programming, *Steve Oualline*, O'Reilly Media, Inc., 3rd edition, 1997)

Briefer than Kochan's textbook, still comprehensive.



Effective C: An Introduction to Professional C Programming, *Robert C. Seacord*, William Pollock, 2020.

Great if you already known some of C syntax and like to improve your skill further.



Fluent C, Principles, Practices, and Patterns, *Christopher Preschern*, O'Reilly Media, Inc., 2022.

Suitable if you like to know more about coding practices.



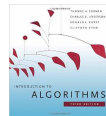
21st Century C: C Tips from the New School, *Ben Klemens*, O'Reilly Media, 2012.



Další zdroje



Introduction to Algorithms, 3rd Edition, *Cormen, Leiserson, Rivest, and Stein*, The MIT Press, 2009, ISBN 978-0262033848



Algorithms, 4th Edition, *Robert Sedgewick, Kevin Wayne*, Addison-Wesley, 2011, ISBN 978-0321573513



The C++ Programming Language, 4th Edition (C++11), *Bjarne Stroustrup*, Addison-Wesley, 2013, ISBN 978-0321563842



Způsob výuky programování v B0B36PRP

- Naší snahou je vybudovat zkušenost a rozvinout dovednost programování.
 - Programování vs. algoritmizace;
 - Programování je „řemeslo“, jak správně implementovat nějaký algoritmus.
 - **Jen funkční nestačí - program musí být i správně!** *Očekávaný vstup vs. co všechno může uživatel na vstup zadat.*
- Studijní zátěž je proto rozložena do výukové části semestru zaměřené na **formativní výuku**.
 - Úkoly na cvičení a termíny domácích úkolů.
- Systematické rozvíjení dovednosti programování v průběhu semestru je zásadní.

Na začátku semestru čas pro čtení učebnice!
- Formativní zpětná vazba v semestru, ale **zkouška je certifikačního typu** (ano/ne).
- Vědět a umět použít (nikoliv “slepotat”). *Nezávislost na našeptávači!*
 - Začínáme relativně jednoduchými úlohami k osvojení programovacích konstruktů a způsobu organizace zdrojového kódu. *Přehledný kód a schopnost se efektivně orientovat v kódu!*
 - *Úkoly jdou vždy realizovat s tím, co si řekneme na přednášce/cvičení.*

Řešení s pokročilejšími konstrukty může být elegantnější(kratší), nemusí však dodat potřebný vhled.
 - V prvních přednáškách pokrýváme nezbytné znalosti, které jsou dále prohlubovány.
 - Cvičení dopňují přednášky a dávají více prostoru pro praktické osvojení problematiky.
- Můžete volit praktický způsob vstřebávání znalosti programování z příkladů, který je vhodný doplnit **teoretickou přípravou z učebnic(e)**.

Přednášky – zimní semestr (ZS) akademického roku 2025/2026

- Harmonogram akademického roku 2025/2026.

<https://www.fel.cvut.cz/cz/education/harmonogram2526.html>

- Přednášky

- Karlovo náměstí, místnost KN:E-107, čtvrtek, 12:45–14:15.

- 14 výukových týdnů.

13+1 přednášek

- Státní svátek – 17.11.2025 (pondělí).

- Konzultace: čtvrtek 14:30–16:00 - KN:E-328.

Po přednášce, případně po domluvě.

Přehled přednášek

- 01 - Informace o předmětu, Úvod do programování
- 02 - Programování v C
- 03 - Řídící struktury, výrazy a funkce
- 04 - Pole, ukazatel, textový řetězec
- 05 - Práce s textem, ukazatele a volání funkcí
- 06 - Struktury a uniony.
- 07 - Paměťové třídy, standardní knihovny C. Rekurse. (**Základní vlastnosti jazyka C probrány.**)

S. G. Kochan: kapitoly 1–3

S. G. Kochan: kapitoly 2–5 a část 6

S. G. Kochan: kapitoly 4–6 a 12

S. G. Kochan: kapitoly 7, 10 a 11

S. G. Kochan: kapitoly 7, 8 a 11

S. G. Kochan: kapitoly 9, 14, 17

S. G. Kochan: kapitola 16 a Appendix B

- 08 - Spojové struktury
- 09 - Abstraktní datový typ (ADT) - zásobník, fronta, prioritní fronta
- 10 - Stromy
- 11 - Prioritní fronta, halda. Příklad použití při hledání nejkratší cesty v grafu
- 12 - Binární soubory
- 13 - Přesnost výpočtu
- 14 - Rezerva – *Dotazy, informace ke zkoušce*

Přednáška není jen prezentace slidů – interakce, řešení dotazů a diskuse.

Podklady k přednášce jsou k dispozici před přednáškou podobně jako **učebnice**.

Domácí úkoly a další úlohy

- Samostatná práce s cílem osvojit si praktické zkušenosti.
- Jednotné zadání na přednášce a jednotný termín odevzdání.
- Odevzdání domácích úkolů prostřednictvím BRUTE.

<https://cw.felk.cvut.cz/brute>

- Nahrání (upload) archivu s nezbytnými zdrojovými soubory.
- Ověření správnosti implementace automatickými testy.
- Penalizace za překročení počtu uploadů.

Odevzdávejte funkční kódy, nikoliv „pouze“ kódy, které projdou testy!

- Detekce plagiátů

*Cílem řešení úkolů je **získat vlastní zkušenost!***

- Úkoly jsou navrhovány tak, aby byly stihnutelné. Plánujte a hlídejte si čas, včas konzultujte.
- Klíčem k úspěšnému dokončení předmětu je samostatná práce a osvojení si technik a znalostí

Průběžná práce a řešení úkolů!

- Pokud něčemu nerozumíte, **ptejte se!**

Pokud chybujete, tak se učíte, pokud nechybujete, tak už to umíte!

Přehled domácích úkolů

- Domácí úkoly s povinným, **volitelným**, případně bonusovým zadáním. 39 h, bonus +19 h
<https://cw.fel.cvut.cz/wiki/courses/b0b36prp/hw/start>
 - 0. HW 00 - První program 1 h
 - 1. HW 01 - Načítání vstupu, výpočet a výstup 2 h
Seznámení se s prostředím, psaním programů, jejich laděním, testováním a odevzdáváním. ~ 20–40 h
 - 2. HW 02 - První cyklus (**Kontrola přehlednosti kódu** – až -100% z dosažených bodů) 2 h
 - 3. HW 03 - Kreslení (ASCII art) (**Kontrola kódu** – až -100%) 4 h
 - 4. HW 04 - Četnost výskytu znaků (**Kontrola kódu** – až -100%) 4 h
 - 5. HW 05 - Caesarova šifra (**Kontrola kódu** – až -100%) 8 h
 - 6. HW 06 - Maticové počty (**Kontrola kódu** – až -100%) 8 h; bonus +7 h
 - 7. HW 07 - Hledání textu v souborech 5 h
 - 8. HW 08 - Kruhová fronta v poli - *Dynamicky linkovaná knihovna* 5 h
 - HW 10B - **Integrace** načítání grafu a prioritní fronta v úloze hledání nejkratších cest
 - 11. přednáška, soutěž na extra body +12 h
- Podmínkou zápočtu je úspěšné odevzdání všech povinných domácích úkolů.
- **Zadání a termíny jsou známy, sledujte svůj postup a plánujte si čas!**
- Odevzdání volitelného zadání je doporučeno (není částečné odevzdání).

Celkové body za povinné zadání **25b**, volitelné zadání **15b**, bonusové **10b+**

Odevzdávání domácích úkolů – BRUTE

■ BRUTE – Bundle for Reservation, Uploading, Testing and Evaluation

- Formální kontrola – kompilace programu.
- Testování funkčnosti – **kontrola výstupu pro daný vstup**.
 - Veřejné vstupy a odpovídající výstupy / neveřejné vstupy.
- Před uploadem programu si program otestujete sami.
 - S využitím dostupných vstupů a výstupů.
 - Vytvoření vlastních vstupů a laděním programu.
 - Vytvoření vstupů **přiloženým generátorem vstupů**.
 - Ověření výstupu **přiloženým testovacím nebo referenčním programem**.
- Porozumění kódu a kontrola možných stavů.
 - **Pro každý řádek byste měli být schopni odpovědět proč tam je a co dělá!**
 - Pro **každou funkci nebo načtení vstupu** od uživatele analyzujte možné vstupní hodnoty nebo **návratové hodnoty funkcí**!
 - Pokud je z hlediska funkčnosti vstup nebo návratová hodnota zásadní, **proved'te kontrolu vstupu a/nebo příslušnou akci**, jako je vypsání hlášení a ukončení programu.

Např. očekávaný vstup je číslo a uživatel zadá něco jiného.

Úkoly a BRUTE

- Cílem úkolů není získat body (ty jsou motivace).
- Výukový cíl není ani odevzdání implementace, která projde BRUTE testy.

Je to jeden z prostředků ověření základní funkčnosti programu.

- **Cíl je osvojit si programovací návyky, programovat samostatně funkční programy správně.**
- BRUTE je nástroj průběžné kontroly postupu a získávání znalostí.
- Úkoly jsou o **postupném získání zkušeností** s konkrétními konstrukty.
- Úkoly míří na začátečníky, jsou proto učebnicové, dobře popsane a zvládnutelné.
Velmi dobře i jazykovými modely LLM.
- Úkoly mají relativní obtížnost velmi podobnou.
 - Je důležité postupně samostatně řešit jednotlivé úkoly a osvojovat si dílčí dovednosti.
Absolutně jsou úlohy postupně náročnější a náročnější!
- Netrapte se s řešením příliš dlouho sami, ptejte se cvičících, přednášce, **konzultaci**.
- Úlohy HW02–HW06 budou manuálně kontrolovány na správnost a přehlednost kódu.
 - Zaměřeno na konzistenci, čitelnost, a **dekompozici** (rozdělení do funkcí).
Z hlediska tréninku a učení, i zdánlivě triviální program se snažte rozumně rozdělit na více funkcí.
 - *Motivace je netrávit příliš mnoho času implementací bez výrazného postupu.*

Hodnocení předmětu

Zdroj bodů	Maximum bodů	Přípustné minimum bodů
Domácí úkoly	40	} 35
Bonusové úkoly	10 ⁺	
Testy, kvízy, aktivita	15	
Písemný zkouškový test	20	[†] 10
Implementační zkouška	20	10
Součet	100 ⁺ bodů	

[†]V případě neúspěšné implementace, lze opakovat pouze implementační část zkoušky pokud byl test hodnocen 13 více body.

- **Zápočet**: alespoň 35 bodů a úspěšně odevzdané všechny povinné úkoly **do 10.1.2026@23:59 CET!**
- Předmět lze úspěšně ukončit **zápočtem a zkouškou**.
Zkouškové termíny (KOS) a <https://cw.fel.cvut.cz/wiki/courses/b0b36prp/start>.
- Test a písemná zkouška – krátké stručné odpovědi prokazující porozumění problematice.
<https://cw.fel.cvut.cz/wiki/courses/b0b36prp/resources/test>
- Implementační zkouška – **certifikační** typ zkoušky, samostatně napsat krátký program.
<https://cw.fel.cvut.cz/wiki/courses/b0b36prp/resources/exam>

Klasifikace předmětu

Klasifikace	Bodové rozmezí	Hodnocení	Slovní hodnocení
A	≥ 90	1	výborně
B	80–89	1,5	velmi dobře
C	70–79	2	dobře
D	60–69	2,5	uspokojivě
E	50–59	3	dostatečně
F	< 50	4	nedostatečně

- Včasné odevzdáním všech domácích úkolů s povinným a **volitelným** zadáním (40 bodů).
- Bonusová úloha a bonusové odevzdání HW10 (10⁺ bodů).
- Testy, kvízy, aktivita (15 bodů).
- Písemná zkouška (20 bodů) . 15 a více bodů je velmi slušný výsledek!
- Implementační zkouška (20 bodů).
- **95 bodů** a více (A – výborně), **76 bodů** (C – dobře) – (20% ztráta).
- Body jsou indikátorem průběžných výsledků.

Komunikace související s PRP

- Obracejte se na své cvičící dle cvičení, na které chodíte (jste přihlášení).

Případně na prp-teachers@fel.cvut.cz

- Komunikovat můžete elektronickou poštou (e-mail).
 - Pište ze své **fakultní adresy** (odesílatel).
 - **Do předmětu zprávy uvádějte zkratku předmětu PRP.**
 - Kopii zprávy (Cc) posílejte též příslušnému vedoucímu cvičení (dle studijního programu).
 - V případě zásadních problému (např. týkajících se zápočtu) uvádějte do Cc též přednášejícího.
- *Zkuste se vyvarovat snaze řešit problémy pohledem do vašich notebooků v krátké pauze mezi cvičeními. Pošlete raději e-mail, třeba i jen s odkazem na kód v BRUTE, nebo v příloze.*

Podobně jako vy, i cvičící potřebují nějaký čas a případně komfort vlastního prostředí. Zpětnou vazbu dostane, případně pošlete předem, nebo využijte konzultace.

Vývojové prostředí

- Počítačové laboratoře - Ubuntu se síťovým bootováním a domovskými adresáři (NFS v4).
Přenos a synchronizace souborů – ownCloud, SSH, gdrive, sharepoint
- Doporučený operační systém - Ubuntu-based, Pop OS!, Win s WSL(2) ideální s nativní VS Code.
Přímočará instalace potřebných programů.
- Překladače **gcc** a **clang**, sestavení **make** (GNU make). <https://gcc.gnu.org> a <http://clang.llvm.org>
- **Visual Studio Code** (VSC) - <https://code.visualstudio.com/>
- Editor – **gedit**, atom, **sublime**, vim – <https://atom.io/>, <http://www.sublimetext.com/>
<http://www.root.cz/clanky/textovy-editor-vim-jako-ide>
Pokud programovat umíte, investuje čas do efektivního ovládání editoru, např. **vim**.
- C/C++ vývojové prostředí – WARNING: Do Not **Rely** on an IDE.
<http://c.learncodethehardway.org/book/ex0.html>
 - **CLion**, NetBeans 8.0 (C/C++), Eclipse-CDT – <https://www.jetbrains.com/clion>
 - **Geany**, Code::Blocks, CodeLite <https://www.geany.org/>, <http://www.codeblocks.org>, <http://codelite.org>
 - **Nejdříve porozumějte principům**, nakonfigurujte nástroje a programování zefektivněte.
- Odevzdávání domácích úkolů BRUTE <https://cw.felk.cvut.cz/brute>.
BRUTE – Bundle for Reservation, Uploading, Testing and Evaluation.

Služby akademické sítě – FEL, ČVUT

- <http://www.fel.cvut.cz/cz/user-info/index.html>
- Diskové úložiště ownCloud – <https://owncloud.cesnet.cz>
- Zasílání velkých souborů – <https://filesender.cesnet.cz>
- Rozvrh a termíny – FEL Portal – <https://portal.fel.cvut.cz>
- FEL Google Account – autentizovaný přístup do Google Apps for Education
Více viz <http://google-apps.fel.cvut.cz/>.
- Gitlab FEL – <https://gitlab.fel.cvut.cz/>
- Přístup k informačním zdrojům (IEEE Xplore, ACM, Science Direct, Springer Link).
- Akademické a kampusové licence. <https://download.cvut.cz>
- Národní Gridová Infrastruktura MetaCentrum. <http://www.metacentrum.cz/cs/index.html>
- RCI Cluster. <https://login.rci.cvut.cz>