XXX

Co to celé stojí?

Po této kapitole přestaneš u ceníku pokrčit rameny a začneš počítat. Kontextové okno uvidíš jako rozpočet, ne jako sklad; poznáš, proč tě dlouhý kontext stojí víc, než by odpovídalo jeho délce, a spočítáš si bod zvratu mezi paušálem a platbou podle spotřeby dřív, než ho odhadneš.

> *Context engineering je jemné umění a věda, jak naplnit kontextové okno právě tou správnou informací pro další krok.*
>
> — Andrej Karpathy, *„...context engineering is the delicate art and science of filling the context window with just the right information for the next step.“ · X (Twitter), 25. 6. 2025*

Začnu dvěma účty, protože jsou poctivější než jakékoli dějiny — a oba mi přišly ve stejném týdnu.

Ten první nebyl faktura, byl to nekrolog jedné zvyklosti. Osmnáctého června 2026 Google oznámil, že ruší free tier svého Gemini CLI — ten „industry-leading“ plán, o kterém se rok psalo jako o jediném nástroji, který je *doopravdy* zdarma: tisíc dotazů denně, bez kreditky, bez háčku. Zvykl jsem si na něj jako na kohoutek s vodou. A pak, přesně ve chvíli, kdy si na něj zvykla i zbytek světa, kohoutek zavřeli a nahradili ho platformou, u níž se ceny teprve doplňují. Zůstal návyk a účet. Nebyla to nehoda ani zlá vůle — byl to plán tak starý, že ho znají v učebnicích marketingu pod jménem, ke kterému se za chvíli dostaneme.

Pramen: Google Developers Blog, oznámení k 18. 6. 2026 — zrušení free/Pro/Ultra tierů Gemini CLI, přechod na platformu Antigravity. Konkrétní kvóty a ceny Antigravity zde neuvádíme, dokud je Google oficiálně nepotvrdí.

Ten druhý účet byl skutečná faktura. Po týdnu, kdy jsem vibe codingem hnal agenta přes celý repozitář — „projdi to celé, oprav to, otestuj to, a mimochodem mi vysvětli, jak to funguje“ —, mi dorazilo vyúčtování spotřeby. Číslo, kterému jsem odmítal věřit, dokud jsem si ho sám nerozpočítal na tokeny. Zjistil jsem, že jsem každý den nechal agenta znovu a znovu přečíst tentýž půlmilion tokenů kódu, aby udělal jeden malý zásah. Platil jsem za to, že mu pokaždé naskládám na stůl celou knihovnu, i když potřeboval jednu stránku.

Tahle kapitola je o obou účtech naráz, protože jsou to dvě strany jedné mince. Ceny nástrojů se řídí ekonomikou, kterou nevymysleli inženýři, ale marketéři — a poznat ji je totéž řemeslo jako poznat, že

model si vymýšlí: obojí je otázka *jak bych poznal, že se pletu?* přenesená z pravdy na peníze. A ta druhá, technická polovina — proč mě zrovna dlouhý kontext stál tolik — má kořeny přesně tam, kde jsme skončili v kapitole patnácté: u pracovního stolu, který je konečný, placený a kvadraticky drahý na roztahování.

Pracovní stůl je rozpočet, ne sklad

Vezmi si zpátky metaforu z kapitoly patnácté a otoč ji o čtvrt otáčky. Řekli jsme tehdy: kontextové okno je *pracovní stůl, ne knihovna* — model ví jen to, co mu na stole zrovna leží. Teď k té větě přidej cenovku. Každý kousek papíru, který na stůl položíš, něco stojí; a když ho tam necháš ležet přes celý den, platíš za něj pokaždé znovu. Stůl není jen konečný. Je to *rozpočet*.

To je celá batolecí lekce téhle kapitoly, a stojí za to ji říct pomalu, protože z ní plyne všechno ostatní. Model neúčtuje za to, co *umí* — to, co se naučil při tréninku, nese v parametrech a máš to zaplacené v paušálu nebo v ceně za token bez ohledu na obsah. Model účtuje za to, co mu *položíš na stůl* a co ti z něj *sebere zpátky*. Vstup, který mu naskládáš (tvůj dotaz, přiložené soubory, celá dosavadní konverzace), a výstup, který ti vrátí. Obojí se měří ve stejné měně, ve které kapitola patnáctá měřila překvapení: v *tokenech*, těch kouscích mezi písmenem a slovem.

A protože se všechno počítá v tokenech, dá se to spočítat. To je dobrá zpráva, kterou si z celé kapitoly odnes: ekonomika téhle práce *není* mlha. Je to násobení, které zvládne kdokoli, kdo umí vynásobit cenu za tisíc tokenů počtem tokenů — a přesně tohle násobení dělí lidi, kteří vědí, co platí, od těch, co jen krčí rameny nad fakturou.

→ kap. 15: token — *kousek textu, na který si model rozseká vstup i výstup; ne slovo, ne písmeno. Časté slovo jeden token, vzácné se drolí na drť. Čeština stojí zhruba dvakrát víc tokenů než tentýž text anglicky — a účtuje se za tokeny, ne za znaky.*

Cena je krutě asymetrická — a výstup je drahý

TEENAGER · MECHANISMUS

Tři čísla, která musíš znát, a jedno, které tě překvapí. Cena se dnes udává za *milion tokenů* (píše se MTok), zvlášť za vstup a zvlášť za výstup. Řádové ceny modelů Anthropic k červenci 2026 (za milion tokenů):

model	vstup	výstup	
Haiku 4.5	1 $	5 $	← malý, rychlý, levný
Sonnet 4.6	3 $	15 $	← střední, pracovní kůň
Opus 4.8	5 $	25 $	← velký, nejchytřejší

Všimni si toho, co se opakuje ve všech třech řádcích: *výstup stojí pětkrát víc než vstup.* To není náhoda — je to fyzika stroje z kapitoly patnácté. Vstup model přečte naráz, jedním průchodem. Výstup musí vyrábět token po tokenu, a při každém dalším tokenu znovu přečte všechno, co už napsal. Proto je psaní dražší než čtení. A proto se vyplatí ptát se tak, aby odpověď byla *hutná*, ne ukecaná: každý zbytečný odstavec platíš nejdražší sazbou.

Ceny řádové, stav k červenci 2026, a stárnou rychle — ověřuj v ceníku. Batch (dávkové zpracování) je zhruba o polovinu levnější; cachování *(uložení stejného vstupu, aby se nečetl znovu) srazí cenu opakovaného vstupu až o 90 %. Sonnet 5 vyšel 30. 6. 2026 s úvodní cenou 2 $/10 $ do konce srpna, pak standardní 3 $/15 $. Prameny: Anthropic ceník; CloudZero,* Claude API Pricing 2026.

Teď to spojme s tím účtem za týden vibe codingu. Řekněme, že agent při každém zásahu přečte 200 tisíc tokenů kontextu (repozitář na stole) a napíše 5 tisíc tokenů odpovědi. Se Sonnetem to je 0,2 × 3 $ + 0,005 × 15 $ = 0,675 $ za jeden zásah. Zdánlivě nic. Jenže při vibe codingu neuděláš jeden zásah — uděláš jich za den klidně sto. To je 67 $ za den, přes tisíc dolarů za měsíc. A drtivá většina té částky je vstup, který jsem tam nechával ležet zbytečně: kdybych agentovi dal na stůl místo celého repozitáře jen tři relevantní soubory, kontext spadne na dvacetinu a s ním i faktura.

A tady je ta úleva: účet za tokeny je jediná faktura, kterou umíš *vysvětlit do posledního centu*. Není v ní nic skrytého — je to počet tokenů krát cena. Kdo tomu rozumí, přestane platit za návyk a začne platit za práci. Karpathyho odhad, že zhruba devět desetin útraty za AI kódování je promrhaných na zbytečném kontextu, není nadávka — je to pozvánka: většina té faktury je úklid stolu, který jsi neudělal.

Kaskáda: malý model doma, velký v oblaku

Z asymetrie cen plyne první velké řemeslo téhle kapitoly, a je to týž trik, jaký dělá dobrý šéf s týmem: neposílej na každý úkol toho nejdražšího člověka.

TEENAGER · MECHANISMUS

Kaskáda malý → velký. Ne každý dotaz potřebuje nejchytřejší model. Rutinu — přeformátuj tenhle text, roztřiď tyhle řádky, napiš commit zprávu — zvládne malý a levný model (Haiku, nebo rovnou model běžící na tvém notebooku) za zlomek ceny. Těžké přemýšlení — navrhni architekturu, najdi tu zákeřnou chybu — pošli velkému modelu do oblaku. Kaskáda znamená: *zkus to nejdřív malým, a k velkému postup jen tehdy, když malý nestačí.*

V praxi to vypadá takhle — v Claude Code přepneš model podle váhy úkolu:

```
/model haiku      # rutina: rychle a levně
/model sonnet     # běžná práce: pracovní kůň
/model opus       # těžké přemýšlení: nejdražší sazba
```

Rozdíl v ceně mezi Haiku a Opusem je pětinásobek na vstupu a pětinásobek na výstupu. Poslat rutinu Opusovi je jako vozit rohlíky kamionem.

→ kap. 29 (bezpečnost): lokální model má i druhou přednost — data z tvého stroje neopustí. Kaskáda „malý-lokální + velký-cloud" tedy neřeší jen cenu, ale i to, co posíláš ven. Nejlevnější a nejbezpečnější token je ten, který nikdy neodejde po drátě.

Kaskáda má i podobu, kterou nevidíš: velcí poskytovatelé sami *kaskádují* uvnitř. Model odměny, cachování, směrování snadných dotazů na levnější hardware — tvůj paušál je průměr přes tohle všechno. Ty děláš zvenčí totéž, co oni uvnitř: snažíš se, aby drahý výpočet dostaly jen úlohy, které ho opravdu potřebují.

> *Účtování po tokenech je nejpoctivější taxametr, jaký kdy vznikl — a zároveň nejzákeřnější, protože jede i tehdy, když se díváš z okna. Agent, kterého jsi pustil „ať si to promyslí", je taxík, co krouží kolem bloku a čeká, jestli si ještě něco nevzpomeneš. Karpathy tomu říká context engineering; taxikář by tomu řekl „nechal jste zapnutý taxametr, pane".*

Zdarma, které vydrží do prvního návyku

Vraťme se k tomu prvnímu účtu — k zavřenému kohoutku. Proč vůbec někdo rozdává nástroj zadarmo, aby ho pak zpoplatnil? Protože to funguje, a má to jméno.

TEENAGER · MECHANISMUS

Penetrační cena a náklady na přechod (switching costs). *Penetrační cena* je záměrně nízká — často nulová — vstupní cena, jejímž cílem není vydělat, ale *obsadit*. Získat uživatele, návyk, tvůj workflow. Jakmile si zvykneš — máš v nástroji nakonfigurované skilly, napojené MCP servery, tým, který v něm pracuje —, vzniknou *náklady na přechod*: cena, kterou bys zaplatil za odchod ke konkurenci. Nejen v penězích: v přeučení, v přenastavení, ve ztracené historii. A přesně o tuhle cenu pak poskytovatel může zvednout svou cenu, aniž odejdeš. Tomu se říká *lock-in* — zámek, jehož klíč držíš ty, ale zámek platí on.

penetrační cena — nízká vstupní cena pro rychlé obsazení trhu. náklady na přechod (switching costs) — cena odchodu ke konkurenci (peníze, čas, přeučení, ztracená data). lock-in — stav, kdy tě náklady na přechod drží u dodavatele, i když zdražil. Prameny: klasika oboru (Shapiro & Varian, Information Rules*, 1999); přehled SaaS praxe 2026.*

Google to nevymyslel. Když v roce 1899 dostali dva právníci od Coca-Coly prakticky zadarmo právo lahvovat colu, a když se pak kola rozdávala po nemocnicích a školách, byla to táž sázka: návyk je levný jen na začátku. Ten, kdo první ochutná zdarma a zvykne si, zaplatí později — a rád, protože odvykání bolí víc než ta cena. Free tier, který byl „industry-leading", vydržel přesně do chvíle, než si na něm lidé zvykli. Pak zmizel a zůstal návyk. To není cynismus, to je učebnice — a poznat učebnici v akci je součást téhož úsudku, kvůli kterému je celá tahle kniha.

Odsud plyne praktické varování, ne rada k paranoie. Když stavíš něco, na čem ti záleží, ptej se u každého nástroje: *kolik mě bude stát odejít?* Nástroj postavený na otevřeném standardu (MCP, git, obyčejné soubory) má náklady na přechod nízké — sbalíš se a jdeš. Nástroj, který tě váže na svůj uzavřený formát, svůj cloud, svou paměť, má je vysoké. Obojí smíš použít; jen to první dělej s otevřenýma očima.

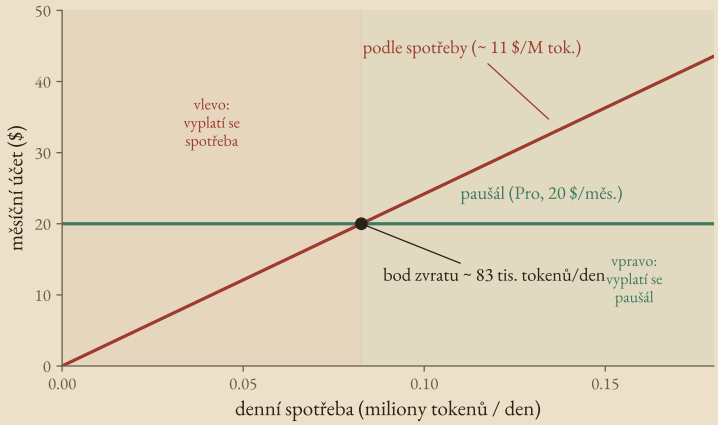
Paušál, nebo podle spotřeby? Spočítej bod zvratu

A teď to, kvůli čemu kapitola stojí, protože tady se hádání mění v počítání. Nástroje ti nabízejí dva světy placení, a rok 2026 je zvláštní tím, že se řada z nich přesouvá z prvního do druhého — GitHub Copilot přešel v červnu 2026 z paušálu na platbu podle spotřeby, Antigravity nahradil rozdaný free tier.

TEENAGER · MECHANISMUS

Dva režimy. *Paušál (flat)*: platíš pevnou částku měsíčně (třeba Claude Pro za 20 $), a v rámci kvóty je jedno, kolik tokenů projedeš. Předvídatelné, ale platíš i za dny, kdy nic neděláš. *Podle spotřeby (usage-based)*: platíš za každý token, co projede. Spravedlivé — kdo nic nedělá, nic neplatí —, ale účet je nepředvídatelný a v aktivním týdnu umí vyletět.

Otázka, kterou si *neháše*, ale spočítáš, zní: *kolik toho denně projedu?* Existuje jedna spotřeba, při níž se oba režimy potkají — *bod zvratu*. Pod ním se vyplatí platit podle spotřeby, nad ním paušál.



Hrdinský graf. Zelená vodorovná = paušál (pevných 20 $). Sanguine stoupající = platba podle spotřeby, roste s tokeny za den. Průsečík = bod zvratu. *Vlevo od něj (málo tokenů) je levnější spotřeba, vpravo paušál. Sklon sanguine přímky je cena za token — dražší model = strmější přímka* = dřívější *bod zvratu.*

Ceny řádové, červenec 2026; efektivní 11 $/M je smíšená sazba vstup+výstup Sonnetu při běžném poměru.

EINSTEIN · MATEMATIKA — přeskočitelné

Spočítej bod zvratu — je to jedna rovnice. Měsíční paušál je pevné číslo F (dolarů). Platba podle spotřeby je cena za token c krát počet tokenů za den t krát počet pracovních dní v měsíci d. Oba režimy stojí stejně tam, kde

$$F = c \cdot t \cdot d.$$

Vyřeš to pro denní spotřebu t, při níž se vyplatí přejít:

$$t_{\text{zvrat}} = \frac{F}{c \cdot d}.$$

Dosaď řádová čísla: paušál $F = 20\,\$$, efektivní cena $c \approx 11\,\$$ za milion tokenů (tedy $0\{,\}000011$ $ za token, smíšený vstup+výstup Sonnetu), $d = 22$ pracovních dní. Vyjde $t_{\text{zvrat}} = 20/(11 \cdot 22) \approx 0\{,\}083$ milionu, tedy zhruba *83 tisíc tokenů denně*. Pod tím se vyplatí spotřeba, nad tím paušál.

83 tisíc tokenů denně je řádově pár dlouhých rozhovorů, nebo pár hodin intenzivního vibe codingu s velkým kontextem. Kdo modelu denně sype celý repozitář, přeletí bod zvratu dopoledne — a paušál (dokud existuje) je pro něj výhra. Kdo se ptá párkrát denně, přeplácí paušál a měl by platit po tokenech.

Všimni si, co ta rovnice říká *mezi řádky*. Bod zvratu není konstanta vesmíru — posouvá se s cenou modelu c. Když přejdeš z Haiku na

Opus, cena za token vyskočí pětinásobně, přímka spotřeby zestrmí a bod zvratu se posune *doleva*: u drahého modelu se ti paušál vyplatí mnohem dřív. A přesně proto poskytovatelé rádi ruší paušály na drahé modely a nechávají je jen na levné — počítají tu rovnici z druhé strany a vědí, kde na ní vyděláš ty a kde oni.

Věž: proč dlouhý kontext stojí víc, než je dlouhý

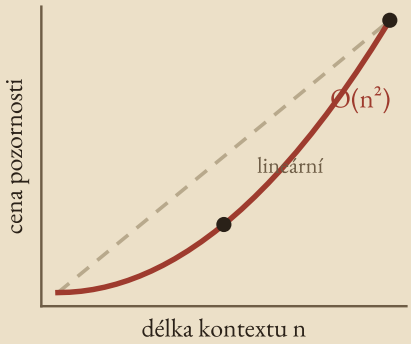
Zbývá vysvětlit tu druhou půlku prvního účtu: proč mě zrovna *dlouhý* kontext stál nepoměrně víc, než by odpovídalo jeho délce. Kdyby dvakrát delší kontext stál dvakrát víc, byla by to spravedlnost. Jenže on stojí *čtyřikrát* víc. Odpověď leží v jediné rovnici z kapitoly patnácté, na kterou teď nasadíme cenovku.

EINSTEIN · MATEMATIKA — přeskočitelné

Kvadratická cena pozornosti, $O(n^2)$. Vzpomeň si na srdce transformeru z kapitoly patnácté: mechanismus pozornosti, kde se každý token zeptá *všech* ostatních, co je pro ně důležité. Tabulka shod všech dotazů se všemi vizitkami má rozměr $n \times n$, kde n je počet tokenů na stole:

$$\text{Pozornost}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V.$$

Ta matice $QK^\top$ má n^2 polí — každý token proti každému. Proto výpočet i paměť rostou s druhou mocninou délky kontextu: zdvojnásob n a práce (a s ní cena za výpočet) se *zečtyřnásobí*. To je ten $O(n^2)$: cena neroste s délkou, ale s *druhou mocninou* délky.



Přerušovaná = naivní očekávání (dvakrát text = dvakrát cena). Sanguine = skutečnost pozornosti, $O(n^2)$. Body: v polovině stolu ($n = 0,5$) je cena jen čtvrtina — a na plném stole čtyřnásobek poloviny. Dlouhý kontext se prodražuje tím rychleji, čím je delší.

EINSTEIN · MATEMATIKA — přeskočitelné

Proč to platíš, i když se cena udává lineárně. Ceník ti účtuje lineárně — tolik dolarů za milion vstupních tokenů, ať jsou v jednom dotazu, nebo ve stovce. Kvadratická cena se schovává jinde: v tom, že poskytovatel musí ten kvadratický výpočet *někde zaplatit*, a promítá ho do ceny za dlouhé kontexty, do pomalejších odpovědí a do stropů (kontextové okno má konečnou velikost právě proto, že n^2 jinak roztrhne paměť). Ty to pocítíš dvakrát: dlouhý kontext je dražší za token *a* pomalejší za token. Odsud plyne řemeslo, které Karpathy pojmenoval — context engineering: *neplň stůl vším; polož na něj právě tu jednu stránku, kterou model pro další krok potřebuje.* Kratší stůl není jen levnější lineárně; je levnější kvadraticky.

→ *kap. 15: kontextové okno* = pracovní stůl, ne knihovna. *Teď víš i cenu úklidu: spadne kvadraticky, ne lineárně. Uklidit stůl na polovinu neušetří polovinu — ušetří tři čtvrtiny.*

Proto ten můj účet za týden. Nechával jsem agentovi na stole celý repozitář — plný stůl, n na maximu, cena na n^2. Kdybych mu dával jen relevantní soubory, nešel bych z ceny na polovinu; šel bych na zlomek. To není spořivost pro spořivost. Je to týž úsudek jako v celé knize: rozhoduje, co položíš na stůl — teď víš, že rozhoduje i o faktuře, a to nelineárně.

Co si odnést

Poskládej to dohromady. Přišly mi dva účty: zavřený kohoutek a faktura za tokeny. Ten první je penetrační cena v akci — zdarma, dokud si nezvykneš, pak lock-in; poznat ji je součást úsudku, ne cynismus. Ten druhý je matematika, kterou umíš spočítat do centu: cena je počet tokenů krát sazba, výstup stojí pětkrát víc než vstup, a kaskáda malý-lokální plus velký-cloud srazí obojí. Kontextové okno není sklad, je to rozpočet — a protože pozornost stojí $O(n^2)$, uklidit stůl na polovinu ušetří tři čtvrtiny, ne polovinu. A volba mezi paušálem a spotřebou není věc názoru: má bod zvratu, jednu rovnici $t_{\text{zvrat}} = F/(c \cdot d)$, kterou buď spočítáš, nebo hádáš.

Příští, poslední kapitola se zeptá na věc, kterou žádná faktura neukáže: *koho poslouchat, když se to všechno — ceny, nástroje, free tiery — mění každý týden?*

DEMO

`lesk-a-bida-vibecodingu.gaussalgo.com/d/kolik-to--stoji`

Kalkulačka tokenů a bodu zvratu: nasyp svůj typický den (kolik dotazů, jak velký kontext, který model) a uvidíš měsíční účet podle spotřeby, čáru paušálu a přesně bod, kde se protnou. Posuň cenu modelu a sleduj, jak bod zvratu ujíždí doleva.

„Jak bych poznal, že se pletu?“

Konkrétní test k této kapitole je jediné číslo: *spočítal jsi bod zvratu, nebo ho hádáš?* Vezmi svůj paušál F, sazbu modelu c, kterým fakticky pracuješ, a odhadni, kolik tokenů denně projedeš (podívej se do `/usage` nebo `/cost` — nástroj ti to řekne přesně). Dosaď do $t_{\text{zvrat}} = F/(c \cdot d)$ a porovnej se svou skutečnou spotřebou. Tvrdím, že většina lidí platí špatný režim: kdo se ptá párkrát denně, přeplácí paušál; kdo sype agentovi celý repozitář, měl by na paušálu klečet a děkovat. Vyjde-li ti, že platíš ten správný režim a přitom jsi to nikdy nespočítal, měl jsi štěstí, ne úsudek — a příště se ceny změní a štěstí ti nepomůže. Kapitola se plete jedině tehdy, ukážeš-li mi svůj účet za tokeny, který *nejde* rozepsat na počet tokenů krát sazba — pak jsi buď našel skrytý poplatek, který má vidět regulátor, nebo, mnohem pravděpodobněji, jsi ten rozpočet nikdy neotevřel.