

Simulace těžby zdrojů

Semestrální práce č.1 z KIV/PGS

Vaším úkolem je naprogramovat vícevláknovou simulaci těžebního procesu a odvozu vytěženého materiálu do skladu za řekou, přes kterou vede přívoz. Cílem semestrální práce je ověřit Vaši schopnost pracovat bezpečně s více vlákny současně a modelovat paralelní zpracování jednoduchých úloh.

Práci implementujte v jazyce Java.

Zadání

Aplikace bude distribuována ve formě spustitelného Java archivu (JAR). Pro spuštění bude stačit zadat následující příkaz:

```
java -jar <cesta k archivu> -i <vstupni soubor> -o <vystupni soubor> -cWorker <int> -tWorker <int> -capLorry <int> -tLorry <int> -capFerry <int>
```

Parametry jsou vysvětleny v následujícím textu.

Vstupní soubor

Vstupem bude textový soubor v kódování ASCII. Soubor vyjadřuje mapu území, na kterém se těží se zachycenými zdroji.

Významové znaky: „ „ (mezera), „X“ a „\n“ (odřádkování, program si musí umět poradit se všemi třemi variantami odřádkování – CR, LF, CRLF).

Mezera značí prázdné místo (není zde co těžít).

X značí zdroj. Zdroj může být osamocen, nebo v **bloku** na jednom řádku – více **X** vedle sebe.

Pro potřeby SP se dvě **X** pod sebou (na různých řádcích) chápou jako **nesousedící**.

Řádky nemusí být stejně dlouhé.

Příklad:

```
x          xxx  x x xxxxxxxxxxxxxxxxxxxx
          x x xxxxxxxxxxxx                x
xx x
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Proces simulace

V procesu vystupuje několik rolí s jasně danými pokyny.

PŘEDÁK – jeho úkolem je před začátkem prací projít mapu a identifikovat pracovní **bloky**. Následně organizuje **DĚLNÍKY** a přiděluje jim práci.

DĚLNÍK – od předáka získává **blok** ke zpracování. Každé políčko **X** zpracovává po náhodně vybranou dobu z intervalu $(0, t_{\text{Worker}}]$. Po zpracování celého bloku odváží výsledek své práce (definovaný počtem zpracovaných políček) k **NÁKLAĐÁKU**. Materiál se nakládá po jednom, každé naložení trvá právě 1s. V daný okamžik může nakládat nejvýše jeden **DĚLNÍK**. **DĚLNÍCI** se ale mohou v nakládání střídat.

Příklad:

Dělník 1 nese 3 zdroje.
Dělník 2 nese 1 zdroj.
Povolená sekvence:
Dělník 1 naloží zdroj.
Dělník 1 naloží zdroj.
Dělník 2 naloží zdroj.
Dělník 1 naloží zdroj.

Po naložení získá od **PŘEDÁKA** další práci a pokračuje. Celkem pracuje zároveň **cWorker** **DĚLNÍKŮ**. **DĚLNÍCI** přicházejí na směnu ráno a pracují až do konce, nejsou v průběhu práce nahrazeni jinými.

NÁKLAĐÁK – v dole se vždy nachází nejvýše jeden **NÁKLAĐÁK**, do kterého lze ukládat materiál. Každý má stejnou kapacitu **capLorry**. Po naplnění kapacity poslední nakládající **DĚLNÍK** **NÁKLAĐÁK** odešle na cestu a připraví nový pro další nakládání.

NÁKLAĐÁK poté po dobu $(0, t_{\text{Lorry}}]$ jede k **PŘÍVOZU**, kde čeká až se naplní kapacita **PŘÍVOZU** **capFerry**. Pro potřeby semestrální práce uvažujte, že počet **NÁKLAĐÁKŮ** bude vždy dělitelný **capFerry** beze zbytku.

Po naplnění kapacity **PŘÍVOZ** přejede na druhou stranu řeky, **NÁKLAĐÁKY** odjedou a **PŘÍVOZ** se vrátí zpět. Pro jednoduchost uvažujeme, že celá tato operace proběhne „okamžitě“.

NÁKLAĐÁKY poté opět po dobu $(0, t_{\text{Lorry}}]$ pojedou na místo určení, kde skončí.

Všechny procesy probíhají **paralelně**.

Koncový stav

- Všechny zdroje jsou vytěženy
- Všechny nákladáky dojely na místo určení
- Počet dovezených zdrojů odpovídá počtu vytěžených zdrojů

Výstupy

Výstupem jsou zapsané zprávy do konzole a do souboru.

Konzole

Do konzole se na začátku simulace přehledně vypíše hodnoty všech zadaných parametrů, a počet nalezených **bloků** a celkový počet **zdrojů**.

Dále se vypíše zpráva pokaždé, když vyjede **PŘÍVOZ**.

Na konci simulace se vypíše:

- počet zdrojů, které každý dělník vytěžil
- celkový počet zdrojů dovezených do cíle

Formát výstupu není zadán, jen by měl být přehledný.

Výstupní soubor

Do výstupního souboru se zapíše seznam všech událostí v programu, ve formátu:

<časové razítko s přesností na milisekundy> <role> <číslo vlákna> <popis>

Zapisované události:

- dokončená analýza vstupního souboru (popis: kolik bloků a kolik zdrojů)
- dokončená těžba zdroje (a jak dlouho to trvalo)
- dokončená těžba bloku (a jak dlouho to trvalo)
- nákladník plný (a jak dlouho čekal na naplnění)
- nákladník dojel k přívozu (a jak dlouho to trvalo)
- přívoz vyrazil (a jak dlouho čekal na naplnění)
- nákladník dojel do cíle (a jak dlouho to od přívozu trvalo)

Přesný formát zápisu není zadán, ale počítejte při návrhu s tím, že by měl být výstup snadno strojově čitelný (tzn. Zpracovatelný jiným programem).

Ostatní (nutné) podmínky

Práce probíhají paralelně a veškeré kritické sekce jsou validně ošetřeny.

Dokumentace bude mít max 1 stranu textu (obrázky se nepočítají), s popisem kritických sekcí v programu.

Vývojovým (či jiným, např. UML) diagramem popište činnost jednotlivých vláken (obecně doporučuji těmito obrázky začít před implementací). Notace diagramu nebude hodnocena, libovolný pochopitelný obrázek postačí.

Kód bude mít kvalitní objektový návrh a bude dostatečně komentován.

Kód i komentáře budou psány v angličtině (kvalita angličtiny nebude hodnocena).

Nesplnění těchto podmínek povede k vrácení SP k doplnění / nápravě.

Odevzdání práce

Práci odevzdejte ve formátu **zip** (nikoliv rar, 7z, tar apod.!) skrze pro tento účel zřízený portlet na Courseware předmětu (záložka Semestrální práce).

Archiv bude mít následující podobu:

Název: oscislo_pgs_vlakna.zip

/src – zdrojový kód (žádné soubory .class, soubory IDE apod!)

/app.jar – spustitelný soubor

/dokumentace.pdf – soubor s dokumentací.

Rady k implementaci

- Dobře si rozmyslete, které z objektů jsou aktivní a potřebují vlastní vlákno, a které ne.
- Než začnete programovat, identifikujte kritické sekce v programu.
- Pro implementaci logiky vlákna použijte rozhraní Runnable namísto dědění od Thread.
- Myslete na zapouzdření a to, že třída by měla mít právě jednu zodpovědnost.
 - Google:
 - encapsulation
 - single responsibility principle
- využijte konzultační cvičení, ať už máte pocit, že na věc jdete správně nebo ne.
 - Pokud si nevíte rady, kontaktujte cvičícího (danekja@kiv.zcu.cz) i mimo konzultační cvičení