

# IOS – projekt 2 (synchronizace)

Zadání je inspirováno knihou Allen B. Downey: The Little Book of Semaphores

## Popis Úlohy (Building H<sub>2</sub>O)

Molekuly vody jsou vytvářeny ze dvou atomů vodíku a jednoho atomu kyslíku. V systému jsou tři typy procesů: (0) hlavní proces, (1) kyslík a (2) vodík. Po vytvoření procesů se procesy reprezentující kyslíky a vodíky řadí do dvou front—jedna pro kyslíky a druhá pro vodíky. Ze začátku fronty vždy vystoupí jeden kyslík a dva vodíky a vytvoří molekulu. V jednu chvíli je možné vytvářet pouze jednu molekulu. Po jejím vytvoření je prostor uvolněn dalším atomům pro vytvoření další molekuly. Procesy, které vytvořily molekulu následně končí. Ve chvíli, kdy již není k dispozici dostatek atomů kyslíku nebo vodíku pro další molekulu (a ani žádné další již nebudou hlavním procesem vytvořeny) jsou všechny zbývající atomy kyslíku a vodíku uvolněny z front a procesy jsou ukončeny.

## Podrobná specifikace úlohy

### Spuštění:

\$ ./proj2 NO NH TI TB

- NO: Počet kyslíků
- NH: Počet vodíků
- TI: Maximální čas milisekundách, po který atom kyslíku/vodíku po svém vytvoření čeká, než se zařadí do fronty na vytváření molekul.  $0 \leq TI \leq 1000$
- TB: Maximální čas v milisekundách nutný pro vytvoření jedné molekuly.  $0 \leq TB \leq 1000$

### Chybové stavy:

- Pokud některý ze vstupů nebude odpovídat očekávanému formátu nebo bude mimo povolený rozsah, program vytiskne chybové hlášení na standardní chybový výstup, uvolní všechny dosud alokované zdroje a ukončí se s kódem (exit code) 1.
- Pokud selže některá z operací se semaforey, nebo sdílenou pamětí, postupujte stejně--program vytiskne chybové hlášení na standardní chybový výstup, uvolní všechny dosud alokované zdroje a ukončí se s kódem (exit code) 1.

### Implementační detaily:

- Každý proces vykonává své akce a současně zapisuje informace o akcích do souboru s názvem proj2.out. Součástí výstupních informací o akci je pořadové číslo A prováděné akce (viz popis výstupů). Akce se číslují od jedničky.
- Použijte sdílenou paměť pro implementaci čítače akcí a sdílených proměnných nutných pro synchronizaci.
- Použijte semaforey pro synchronizaci procesů.
- Nepoužívejte aktivní čekání (včetně cyklického časového uspání procesu) pro účely synchronizace.
- Pracujte s procesy, ne s vlákny.

## Hlavní proces

- Hlavní proces vytváří ihned po spuštění NO procesů kyslíku a NH procesů vodíku.
- Poté čeká na ukončení všech procesů, které aplikace vytváří. Jakmile jsou tyto procesy ukončeny, ukončí se i hlavní proces s kódem (exit code) 0.

## Proces Kyslík

- Každý kyslík je jednoznačně identifikován číslem  $idO$ ,  $0 < idO \leq NO$
- Po spuštění vypíše: *A: O idO: started*
- Následně čeká pomocí volání `usleep` náhodný čas v intervalu  $<0, TI>$
- Vypíše: *A: O idO: going to queue* a zařadí se do fronty kyslíků na vytváření molekul.
- Ve chvíli, kdy není vytvářena žádná molekula, jsou z čela front uvolněny kyslík a dva vodíky. Příslušný proces po uvolnění vypíše: *A: O idO: creating molecule noM* ( $noM$  je číslo molekuly, ty jsou číslovány postupně od 1).
- Pomocí `usleep` na náhodný čas v intervalu  $<0, TB>$  simuluje dobu vytváření molekuly.
- Po uplynutí času vytváření molekuly informuje vodíky ze stejné molekuly, že je molekula dokončena.
- Vypíše: *A: O idO: molecule noM created* a proces končí.
- Pokud již není k dispozici dostatek vodíků (ani nebudou žádné další vytvořeny/zařazeny do fronty) vypisuje: *A: O idO: not enough H* a proces končí.

## Proces Vodík

- Každý vodík je jednoznačně identifikován číslem  $idH$ ,  $0 < idH \leq NH$
- Po spuštění vypíše: *A: H idH: started*
- Následně čeká pomocí volání `usleep` náhodný čas v intervalu  $<0, TI>$
- Vypíše: *A: H idH: going to queue* a zařadí se do fronty vodíků na vytváření molekul.
- Ve chvíli, kdy není vytvářena žádná molekula, jsou z čela front uvolněny kyslík a dva vodíky. Příslušný proces po uvolnění vypíše: *A: H idH: creating molecule noM* ( $noM$  je číslo molekuly, ty jsou číslovány postupně od 1).
- Následně čeká na zprávu od kyslíku, že je tvorba molekuly dokončena.
- Vypíše: *A: H idH: molecule noM created* a proces končí.
- Pokud již není k dispozici dostatek kyslíků nebo vodíků (ani nebudou žádné další vytvořeny/zařazeny do fronty) vypisuje: *A: H idH: not enough O or H* a proces končí.

## Obecné informace

- Projekt implementujte v jazyce C. Komentujte zdrojové kódy, programujte přehledně. Součástí hodnocení bude i kvalita zdrojového kódu.
- Kontrolujte, zda se všechny procesy ukončují korektně a zda při ukončování správně uvolňujete všechny alokované zdroje.
- Dodržujte syntax zadaných jmen, formát souborů a formát výstupních dat. Použijte základní skript pro ověření korektnosti výstupního formátu (dostupný z webu se zadáním).
- Dotazy k zadání: Veškeré nejasnosti a dotazy řešte pouze prostřednictvím diskuzního fóra k projektu 2.
- Poznámka k testování: Můžete si nasimulovat častější přepínání procesů například vložení krátkého uspání po uvolnění semaforů apod. Pouze pro testovací účely, do finálního řešení nekládejte!

## Překlad

- Pro překlad používejte nástroj make. Součástí odevzdání bude soubor Makefile.
- Překlad se provede příkazem make v adresáři, kde je umístěn soubor Makefile.
- Po překladu vznikne spustitelný soubor se jménem proj2, který bude umístěn ve stejném adresáři jako soubor Makefile
- Spustitelný soubor může být závislý pouze na systémových knihovnách---nesmí předpokládat existenci žádného dalšího studentem vytvořeného souboru (např. spustitelný soubor vodík, konfigurační soubor, dynamická knihovna kyslík, ...).
- Zdrojové kódy překládejte s přepínači -std=gnu99 -Wall -Wextra -Werror -pedantic
- Pokud to vaše řešení vyžaduje, lze přidat další přepínače pro linker (např. kvůli semaforům či sdílené paměti, -pthread, -lrt , . . . ).
- Vaše řešení musí být možné přeložit a spustit na serveru *merlin*.

## Odevzdání

- Součástí odevzdání budou pouze soubory se zdrojovými kódy (\*.c , \*.h ) a soubor Makefile. Tyto soubory zabalte pomocí nástroje zip do archivu s názvem proj2.zip.
- Archiv vytvořte tak, aby po rozbalení byl soubor Makefile umístěn ve stejném adresáři, jako je archiv.
- Archiv proj2.zip odevzdejte prostřednictvím informačního systému—termín Projekt 2.
- Pokud nebude dodržena forma odevzdání nebo projekt nepůjde přeložit, bude projekt hodnocen 0 body.
- Archiv odevzdejte pomocí informačního systému v dostatečném předstihu (odevzdaný soubor můžete před vypršením termínu snadno nahradit jeho novější verzí, kdykoliv budete potřebovat).

## Příklad výstupu

Příklad výstupního souboru proj2.out pro následující příkaz:

```
$ ./proj2 3 5 100 100
```

---

```
1: H 1: started
2: H 3: started
3: O 1: started
4: O 1: going to queue
5: H 2: started
6: H 2: going to queue
7: H 1: going to queue
8: O 3: started
9: O 3: going to queue
10: H 5: started
11: H 4: started
12: O 2: started
13: H 1: creating molecule 1
14: H 4: going to queue
15: O 1: creating molecule 1
16: H 2: creating molecule 1
17: H 5: going to queue
18: H 2: molecule 1 created
19: H 1: molecule 1 created
```

20: O 1: molecule 1 created  
21: H 3: going to queue  
22: O 3: creating molecule 2  
23: O 2: going to queue  
24: H 4: creating molecule 2  
25: H 5: creating molecule 2  
26: O 3: molecule 2 created  
27: H 4: molecule 2 created  
28: H 5: molecule 2 created  
29: H 3: not enough O or H  
30: O 2: not enough H