

Semestrální práce z předmětu 36OSY

Martin Tůma, 3/50

16.04.2005

Obsah

1	Zadání	3
1.1	Zadání	3
1.2	Interpretace zadání	3
2	Implementace	3
2.1	Implementace	3
2.2	Použití	3
3	Závěr	4

1 Zadání

1.1 Zadání

Simulujte průběžný sort. Jsou dány procesy P, S a C. Proces P zasílá náhodná čísla procesu S max. rychlostí čísel za jednotku času. Proces S třídí tyto čísla. rychlostí u komparací za jednotku času a zasílá nejnížší dostupná čísla procesu C (pokud nejsou k dispozici, je proces C uspán). Odeslaná data jsou v procesu S smazána. Na závěr chování pro různě velké hodnoty u a v.

1.2 Interpretace zadání

Program vytvoří 3 procesy – P, S a C. Proces P zasílá procesu S náhodná čísla rychlostí v čísel za sekundu. Proces S tato čísla třídí a rychlostí u zasílá vždy nejnížší z aktuálně dostupných čísel procesu C. Proces C je vždy mezi příchody dvou následujících čísel uspán. V okamžiku, kdy je zcela naplněna posloupnost tříděných čísel v procesu S, je generování čísel v procesu P pozastaveno do okamžiku, než je S schopen přijmout další data. Rychlosti u a v přijímá program jako parametry při spuštění.

2 Implementace

2.1 Implementace

Komunikace mezi procesy je řešena pomocí datových rour. Toto řešení je obzvláště výhodné proto, že přenáší většinu synchronizačních problémů na OS, čímž se velmi snižuje složitost implementace. OS totiž při pokusu zapsat data do „plné“ roury automaticky pozastaví generující proces a naopak, pokud nejsou v rouře k dispozici žádná data, uspí proces čtoucí z roury.

Pro řazení prvků jsem zvolil prioritní frontu, která je implementovaná v samostatné knihovně (*prifo.h*, *prifo.c*).

Provedení řazení v daných intervalech je zajištěno pomocí signálu SIGALRM, který je v intervalech daných požadovanou rychlostí u vyvoláván funkcí *ualarm()* a zpracován příslušnou obslužnou funkcí.

2.2 Použití

```
$ pipesort v u
```

```
v - rychlost generátoru [znaků/s]
```

```
u - rychlost třídiče [znaků/s]
```

Výstup programu je vypisován na standartní výstup ve formě:

```
PROCES: N
```

kde PROCES je jméno procesu (P/C) a N je aktuální odeslané/přijaté číslo.

Program byl testován na OS LINUX a Solaris, Je však pravděpodobné, že půjde přeložit i na jiných UNIXových systémech. K usnadnění překlada je součástí zdrojových kódů i *Makefile* soubor pro automatický překlad pomocí utility *make*.

3 Závěr

Při testování programu jsem nenarazil na žádné nepředvídatelné chování programu pro žádné parametry u a v , lze tedy usuzovat, že splňuje zadání z části „Interpretace zadání“. Jak dalece se však blíží skutečně zamýšlenému zadání lze díky velmi vágnímu zadání jen těžko odhadnout.

Literatura

- [1] *The GNU C Library*
http://www.gnu.org/software/libc/manual/html_node/
- [2] *Programování pod Linuxem pro všechny*
<http://www.root.cz/clanky/programovani-pod-linuxem-pro-vsechny/>
- [3] *Materiály k 36OSY*
<http://service.felk.cvut.cz/courses/X36OSY/>