



B Universitat de Barcelona

Informe pràctica 3

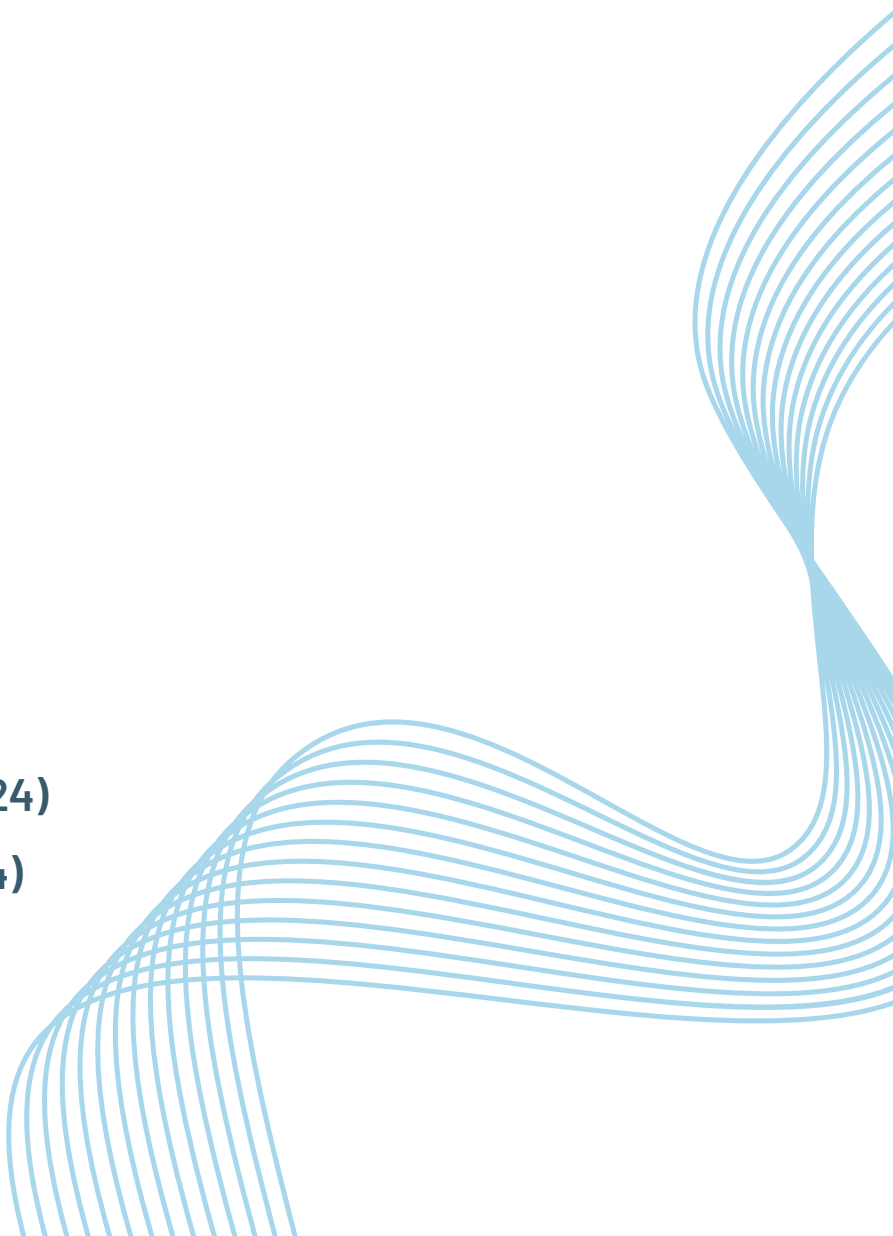
Sistemes Operatius I, 2022

5 de Maig de 2022

fet per:

Oscar de Caralt Roy (Grup 24)

Maria Pérez Pullès (Grup 24)



Índex

Introducció	2
Proves algorisme implementat	2
Preguntes i observacions	3
Conclusions	3

Introducció

Aquesta pràctica se centra a utilitzar alguns dels mètodes que ens ofereix el sistema operatiu per comunicar processos entre si. Se'ns demana que desenvolupem un esquema de productor-consumidor. Aquest paradigma és un exemple clàssic de sincronització de processos. De forma general, en aquest paradigma de programació hi ha diversos processos. Els productors "produeixen" informació i l'escrueixen en un búffer. Amb això se la "entreguen" als consumidors, els quals llegeixen les dades del búffer i realitzen alguna operació sobre aquestes dades. A l'hora d'implementar aquest esquema cal assegurar que el consumidor no intenti agafar dades si el búffer és buit. De manera similar, el productor no pot introduir dades al búffer fins que el consumidor les hagi agafat.

El productor extreu les dades del fitxer data.csv. Es tracta d'un fitxer de text de 99999 línies de dades més la capçalera. El fitxer conté dades sobre viatges en taxi a la ciutat de Nova York. Cada línia del fitxer conté les dades d'un viatge. El productor, de forma iterativa, llegeix cada línia del fitxer i n'extreu les columnes 8 i 9 que indiquen, respectivament, quants passatgers anaven al taxi i quant ha durat cada viatge, en segons. Els valors de la columna 8 seran escrits al fitxer col8.bin, mentre que els valors de la columna 9 seran escrits al fitxer col9.bin. Cada vegada que el productor hagi introduït N valors a cadascun dels fitxers realitzarà una notificació, mitjançant un senyal, als consumidors i aquests llegiran les dades en faran la mitja i ho notificaran al productor.

Proves algorisme implementat

- **Prova 1: Blocs de 20.000 línies (N=20.000)**

```
oslab: /media/sf_S01/P3 bona> gcc -c plantilla.c -o plantilla.o
oslab: /media/sf_S01/P3 bona> gcc plantilla.o -o p3
oslab: /media/sf_S01/P3 bona> ./p3 data.csv 20000
llegint 99999 elements de fd_col8...
Mitjana de passatgers: 2.163002
llegint 99999 elements de fd_col9...
Mitjana del temps de desplaçament: 650.639771 s
EL pare ha acabat el procés
```

- **Prova 1: Blocs de 5.000 línies (N=5.000)**

```
oslab: /media/sf_S01/P3 bona> gcc -c plantilla.c -o plantilla.o
oslab: /media/sf_S01/P3 bona> gcc plantilla.o -o code
oslab: /media/sf_S01/P3 bona> ./code data.csv 5000
llegint 99999 elements de fd_col8...
Mitjana de passatgers: 2.163002
llegint 99999 elements de fd_col9...
Mitjana del temps de desplaçament: 650.639771 s
EL pare ha acabat el procés
```

Preguntes i observacions

- **Atès el context especificat, es poden fer servir les mateixes variables globals (per exemple, sigusr1 i sigusr2) als tres processos per gestionar la recepció dels senyals? O farà falta que cada procés tingui la seva pròpia variable global (per exemple, sigusr_pare, sigusr1_consumidor1, sigusr1_consumidor2 i l'equivalent per a sigusr2)?**
 - Sí, es poden fer servir les mateixes variables globals (per exemple, sigusr1 i sigusr2) als tres processos per gestionar la recepció dels senyals.
- **Podeu raonar per què cal fer servir les crides a sistema i no es poden fer servir les crides a la llibreria d'usuari?**
 - Les crides a sistema són més òptimes per a la comunicació interprocés, ja que usen el búffer del sistema, encanvi les crides a llibreria d'usuari tenen un búffer d'usuari i això les fa poc òptimes per a la comunicació interprocés que és del que tracta la pràctica.

Conclusions

Hem pogut comprovar que el programa funciona correctament per a diferents valors de N.