

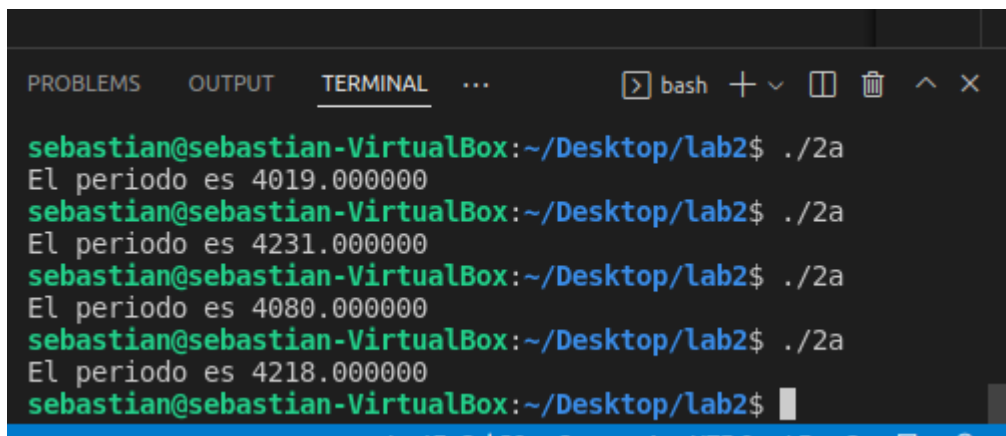
## Laboratorio #2

### Ejercicio 1

- ¿Cuántos procesos se crean en cada uno de los programas?
  - Se crean 16 procesos en cada programa.
- ¿Por qué hay tantos procesos en ambos programas cuando uno tiene cuatro llamadas fork() y el otro sólo tiene una?
  - Ya que se crean 4 forks en ambos entonces tenemos  $2 * 2 * 2 * 2$  lo cual es igual a 16 procesos.

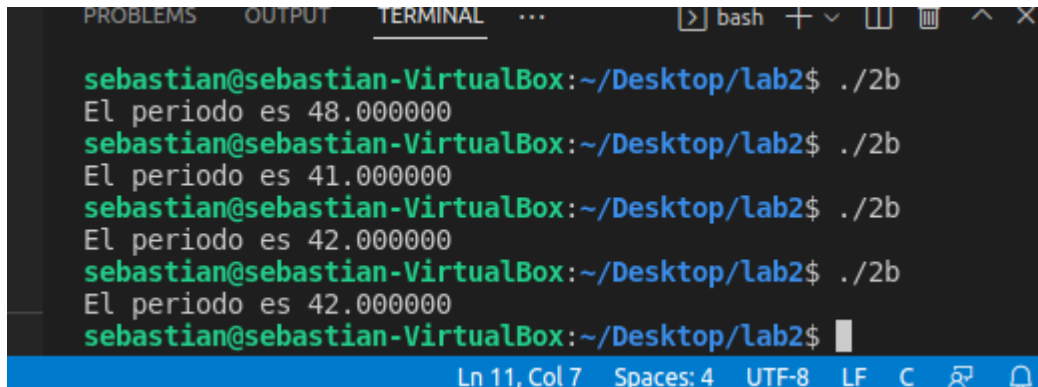
### Ejercicio 2

- 2a



```
sebastian@sebastian-VirtualBox:~/Desktop/lab2$ ./2a
El periodo es 4019.000000
sebastian@sebastian-VirtualBox:~/Desktop/lab2$ ./2a
El periodo es 4231.000000
sebastian@sebastian-VirtualBox:~/Desktop/lab2$ ./2a
El periodo es 4080.000000
sebastian@sebastian-VirtualBox:~/Desktop/lab2$ ./2a
El periodo es 4218.000000
sebastian@sebastian-VirtualBox:~/Desktop/lab2$
```

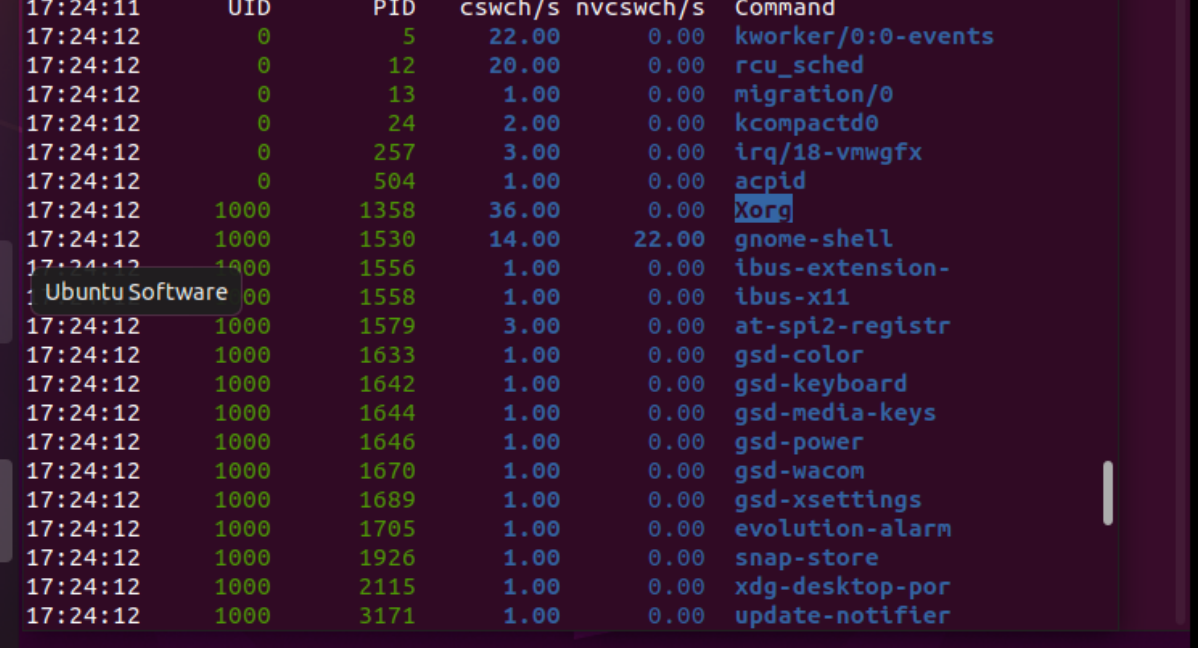
- 2b



```
sebastian@sebastian-VirtualBox:~/Desktop/lab2$ ./2b
El periodo es 48.000000
sebastian@sebastian-VirtualBox:~/Desktop/lab2$ ./2b
El periodo es 41.000000
sebastian@sebastian-VirtualBox:~/Desktop/lab2$ ./2b
El periodo es 42.000000
sebastian@sebastian-VirtualBox:~/Desktop/lab2$ ./2b
El periodo es 42.000000
sebastian@sebastian-VirtualBox:~/Desktop/lab2$
```

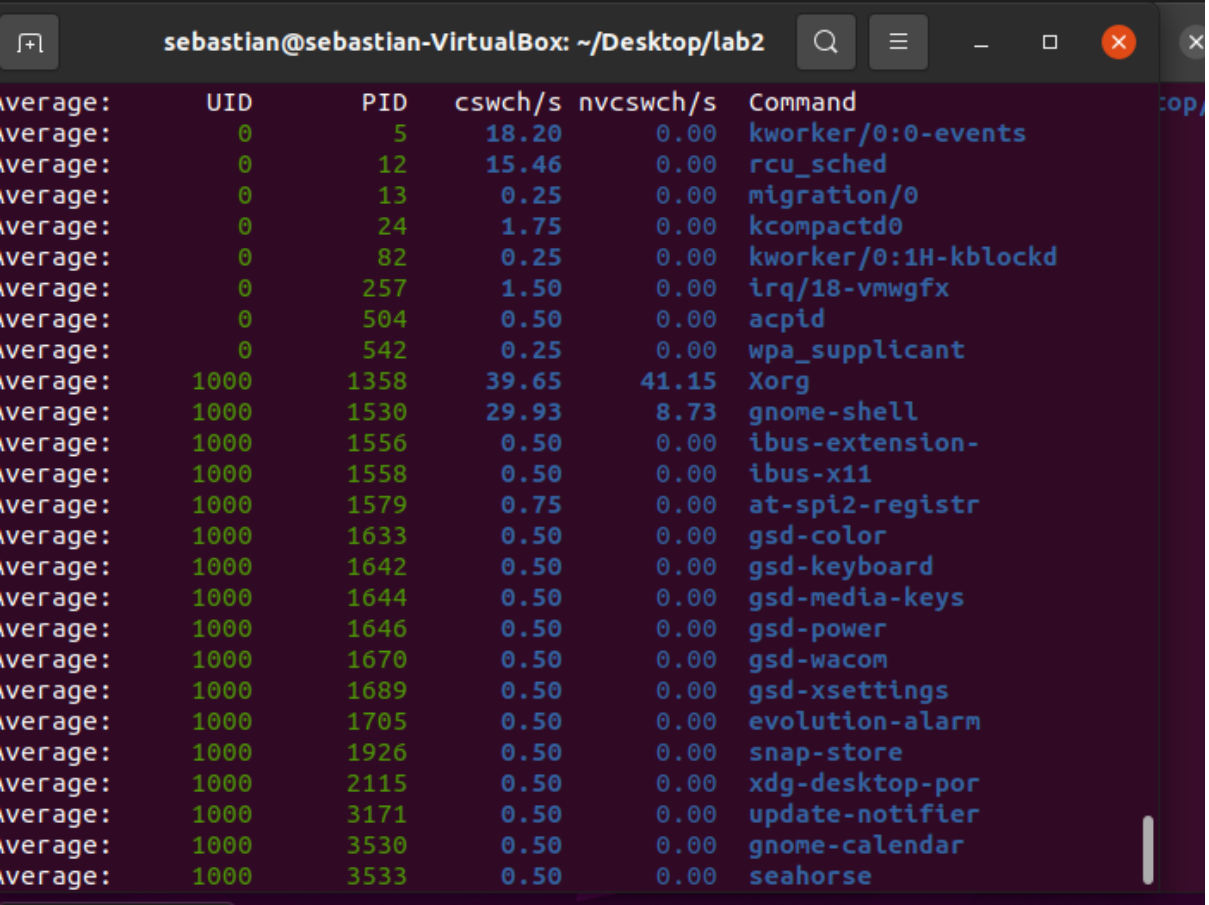
- ¿Cuál, en general, toma tiempos más largos?
  - El primero
- ¿Qué causa la diferencia de tiempo, o por qué se tarda más el que se tarda más?
  - Dado a su manera de ejecutar secuencialmente entonces es más tardado.

### Ejercicio 3



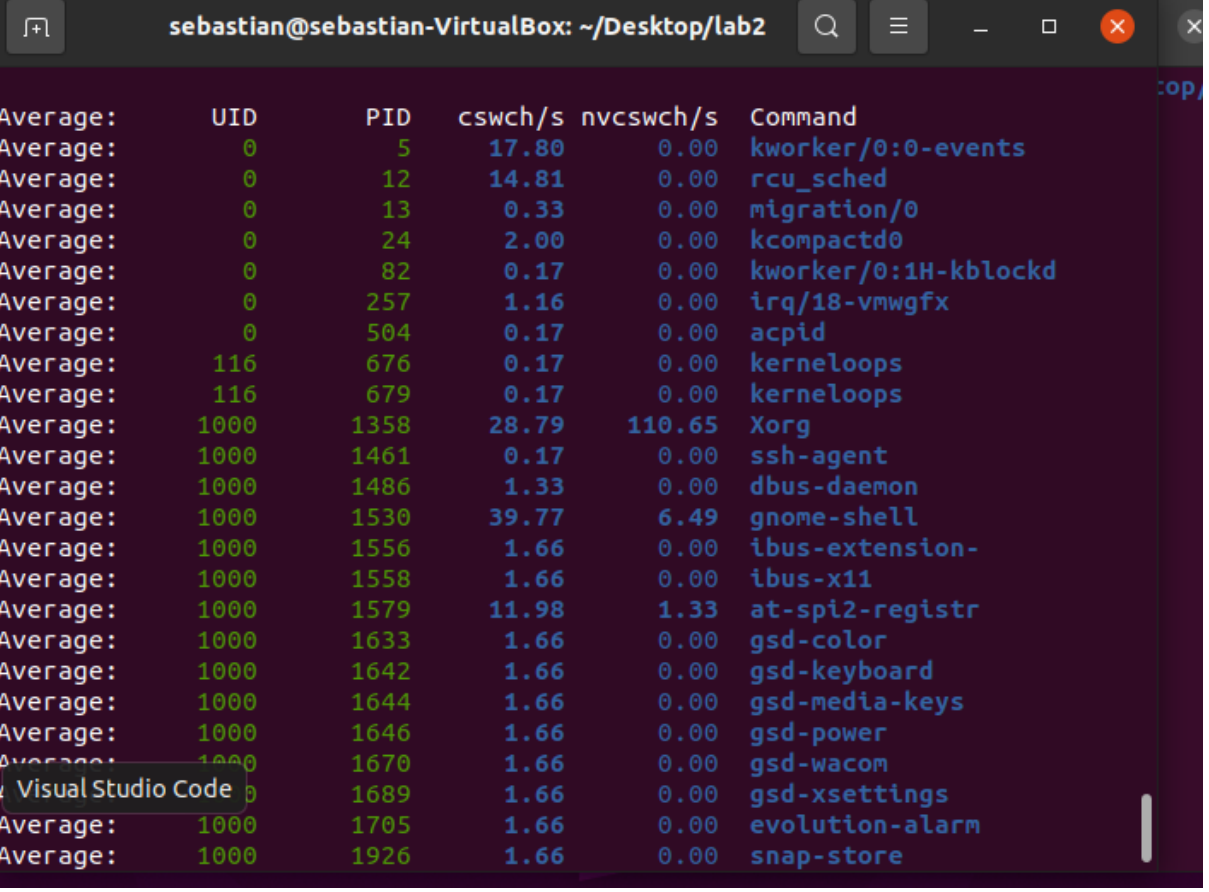
	UID	PID	cswch/s	nvcswh/s	Command
17:24:11	0	5	22.00	0.00	kworker/0:0-events
17:24:12	0	12	20.00	0.00	rcu_sched
17:24:12	0	13	1.00	0.00	migration/0
17:24:12	0	24	2.00	0.00	kcompactd0
17:24:12	0	257	3.00	0.00	irq/18-vmwgfx
17:24:12	0	504	1.00	0.00	acpid
17:24:12	1000	1358	36.00	0.00	Xorg
17:24:12	1000	1530	14.00	22.00	gnome-shell
17:24:12	1000	1556	1.00	0.00	ibus-extension-
17:24:12	1000	1558	1.00	0.00	ibus-x11
17:24:12	1000	1579	3.00	0.00	at-spi2-registr
17:24:12	1000	1633	1.00	0.00	gsd-color
17:24:12	1000	1642	1.00	0.00	gsd-keyboard
17:24:12	1000	1644	1.00	0.00	gsd-media-keys
17:24:12	1000	1646	1.00	0.00	gsd-power
17:24:12	1000	1670	1.00	0.00	gsd-wacom
17:24:12	1000	1689	1.00	0.00	gsd-xsettings
17:24:12	1000	1705	1.00	0.00	evolution-alarm
17:24:12	1000	1926	1.00	0.00	snap-store
17:24:12	1000	2115	1.00	0.00	xdg-desktop-por
17:24:12	1000	3171	1.00	0.00	update-notifier

- ¿Qué tipo de cambios de contexto incrementa notablemente en cada caso, y por qué?
  - El gnome-terminal que es donde se lleva a cabo el proceso ingresado en la terminal.



The image shows a terminal window titled "sebastian@sebastian-VirtualBox: ~/Desktop/lab2". The terminal displays a table of system process statistics. The table has six columns: "average:", "UID", "PID", "cswch/s", "nvcswh/s", and "Command". The data is organized into two groups: processes with UID 0 and processes with UID 1000. The first group includes system processes like kworker, rcu\_sched, migration, kcompactd, kworker/0:1H-kblockd, irq/18-vmwgfx, acpid, and wpa\_supplicant. The second group includes user processes like Xorg, gnome-shell, ibus-extension, ibus-x11, at-spi2-registr, gsd-color, gsd-keyboard, gsd-media-keys, gsd-power, gsd-wacom, gsd-xsettings, evolution-alarm, snap-store, xdg-desktop-por, update-notifier, gnome-calendar, and seahorse.

average:	UID	PID	cswch/s	nvcswh/s	Command
average:	0	5	18.20	0.00	kworker/0:0-events
average:	0	12	15.46	0.00	rcu_sched
average:	0	13	0.25	0.00	migration/0
average:	0	24	1.75	0.00	kcompactd0
average:	0	82	0.25	0.00	kworker/0:1H-kblockd
average:	0	257	1.50	0.00	irq/18-vmwgfx
average:	0	504	0.50	0.00	acpid
average:	0	542	0.25	0.00	wpa_supplicant
average:	1000	1358	39.65	41.15	Xorg
average:	1000	1530	29.93	8.73	gnome-shell
average:	1000	1556	0.50	0.00	ibus-extension-
average:	1000	1558	0.50	0.00	ibus-x11
average:	1000	1579	0.75	0.00	at-spi2-registr
average:	1000	1633	0.50	0.00	gsd-color
average:	1000	1642	0.50	0.00	gsd-keyboard
average:	1000	1644	0.50	0.00	gsd-media-keys
average:	1000	1646	0.50	0.00	gsd-power
average:	1000	1670	0.50	0.00	gsd-wacom
average:	1000	1689	0.50	0.00	gsd-xsettings
average:	1000	1705	0.50	0.00	evolution-alarm
average:	1000	1926	0.50	0.00	snap-store
average:	1000	2115	0.50	0.00	xdg-desktop-por
average:	1000	3171	0.50	0.00	update-notifier
average:	1000	3530	0.50	0.00	gnome-calendar
average:	1000	3533	0.50	0.00	seahorse



Average:	UID	PID	cswch/s	nvcswh/s	Command
Average:	0	5	17.80	0.00	kworker/0:0-events
Average:	0	12	14.81	0.00	rcu_sched
Average:	0	13	0.33	0.00	migration/0
Average:	0	24	2.00	0.00	kcompactd0
Average:	0	82	0.17	0.00	kworker/0:1H-kblockd
Average:	0	257	1.16	0.00	irq/18-vmwgfx
Average:	0	504	0.17	0.00	acpid
Average:	116	676	0.17	0.00	kerneloops
Average:	116	679	0.17	0.00	kerneloops
Average:	1000	1358	28.79	110.65	Xorg
Average:	1000	1461	0.17	0.00	ssh-agent
Average:	1000	1486	1.33	0.00	dbus-daemon
Average:	1000	1530	39.77	6.49	gnome-shell
Average:	1000	1556	1.66	0.00	ibus-extension-
Average:	1000	1558	1.66	0.00	ibus-x11
Average:	1000	1579	11.98	1.33	at-spi2-registr
Average:	1000	1633	1.66	0.00	gsd-color
Average:	1000	1642	1.66	0.00	gsd-keyboard
Average:	1000	1644	1.66	0.00	gsd-media-keys
Average:	1000	1646	1.66	0.00	gsd-power
Average:	1000	1670	1.66	0.00	gsd-wacom
Average:	1000	1689	1.66	0.00	gsd-xsettings
Average:	1000	1705	1.66	0.00	evolution-alarm
Average:	1000	1926	1.66	0.00	snap-store

- ¿Qué diferencia hay en el número y tipo de cambios de contexto de entre programas?
  - El programa de forks tiene un mayor número de cswch/w que es el número de cambios voluntarios de contexto.
- ¿A qué puede atribuir los cambios de contexto voluntarios realizados por sus programas?
  - Los forks que crean procesos nuevos.
- ¿A qué puede atribuir los cambios de contexto involuntarios realizados por sus programas?
  - Los forks compiten por memoria y hacen llamadas al sistema al crear otro proceso por lo que crean cambios involuntarios.
- ¿Por qué el reporte de cambios de contexto para su programa con fork()s muestra cuatro procesos, uno de los cuales reporta cero cambios de contexto?
  - Dado que solo se espera a los hijos y en el padre no se realiza ningún cambio en particular.
- ¿Qué efecto percibe sobre el número de cambios de contexto de cada tipo?
  - Que a medida que aparece un contexto nuevo este duplica las llamadas.

#### Ejercicio 4

```

sttlan@sebastián-virtualbox: ~/Desktop/tao2$ ps -aet
  UID      PID     PPID    C  PRI   NI   ADDR  SZ  WCHAN   TTY          TIME CMD
    0         1         0  0   80    0   - 42203   -    ?         00:00:01 systemd
    0         2         0  0   80    0   -      0   -    ?         00:00:00 kthreadd
    0         3         2  0   60   -20   -      0   -    ?         00:00:00 rcu_gp
    0         4         2  0   60   -20   -      0   -    ?         00:00:00 rcu_par_gp
    0         6         2  0   60   -20   -      0   -    ?         00:00:00 kworker/0:
    0         8         2  0   60   -20   -      0   -    ?         00:00:00 mm_percpu_
    0         9         2  0   80    0   -      0   -    ?         00:00:00 rcu_tasks_
    0        10         2  0   80    0   -      0   -    ?         00:00:00 rcu_tasks_
    0        11         2  0   80    0   -      0   -    ?         00:00:06 ksoftirqd/
    0        12         2  0   80    0   -      0   -    ?         00:00:00 rcu_sched
    0        13         2  0  -40   -   -      0   -    ?         00:00:00 migration/
    0        14         2  0    9   -   -      0   -    ?         00:00:00 idle_injec
    0        16         2  0   80    0   -      0   -    ?         00:00:00 cpuhp/0
    0        17         2  0   80    0   -      0   -    ?         00:00:00 kdevtmpfs
    0        18         2  0   60   -20   -      0   -    ?         00:00:00 netns
    0        19         2  0   60   -20   -      0   -    ?         00:00:00 inet_frag_
    0        20         2  0   80    0   -      0   -    ?         00:00:00 kauditd
    0        21         2  0   80    0   -      0   -    ?         00:00:00 khungtaskd
    0        22         2  0   80    0   -      0   -    ?         00:00:00 oom_reaper
    0        23         2  0   60   -20   -      0   -    ?         00:00:00 writeback
    0        24         2  0   80    0   -      0   -    ?         00:00:00 kcompactd0
    0        25         2  0   85    5   -      0   -    ?         00:00:00 ksm
    0        26         2  0   99   19   -      0   -    ?         00:00:00 khugepaged
    0        72         2  0   60   -20   -      0   -    ?         00:00:00 kintegrity
    0        73         2  0   60   -20   -      0   -    ?         00:00:00 kblockd
    0        74         2  0   60   -20   -      0   -    ?         00:00:00 blkcg_punt

```

```

R 1000 10051 10025 0 80 0 - 9551002 - ? 00:00:04 code
0 S 1000 10074 10025 0 80 0 - 9551095 ep_poll ? 00:00:00 code
0 S 1000 10001 10051 0 80 0 - 3290 do_wai pts/1 00:00:00 bash
1 I 0 12128 4041 0 80 0 - 3289 do_wai pts/2 00:00:00 bash
0 12199 2 0 80 0 - 0 - ? 00:00:00 kworker/u2
0 12299 2 0 80 0 - 0 - ? 00:00:00 kworker/u2
0 12485 2 0 80 0 - 0 - ? 00:00:00 kworker/0:
0 12505 2 0 80 0 - 0 - ? 00:00:00 kworker/0:
0 R 1000 12694 10091 49 80 0 - 591 - pts/1 00:00:02 4a
0 S 1000 12695 12694 1 80 0 - 624 wait_w pts/1 00:00:00 4a
0 S 1000 12723 10051 0 80 0 - 655 do_wai ? 00:00:00 sh
0 S 1000 12724 12723 0 80 0 - 3009 do_wai ? 00:00:00 cpuUsage.s
0 S 1000 12729 12724 0 80 0 - 2655 hrtime ? 00:00:00 sleep

```

- ¿Qué significa la Z y a qué se debe?
  - Es un proceso que está muerto, por eso dice defunct porque ya terminó de ejecutarse. Por lo mismo, ya no utiliza recursos del cpu.
- ¿Qué sucede en la ventana donde ejecutó su programa?
  - Se carga más el output por lo que se puede observar más flujo en el mismo.
- ¿Quién es el padre del proceso que quedó huérfano?
  - El proceso 2 es el padre el thread y sys.