

## Ejercicio 1(10puntos)

- ¿Cuántos procesos se crean en cada uno de los programas?
  - En ambos programas se crearon 15 procesos.

```
root@debian:/home/michael/Desktop/Lab2-S0# gcc -o ej1a ej1a.c
root@debian:/home/michael/Desktop/Lab2-S0# ./ej1a
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
root@debian:/home/michael/Desktop/Lab2-S0# Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
```

```
root@debian:/home/michael/Desktop/Lab2-S0# gcc -o ej1b ej1b.c
root@debian:/home/michael/Desktop/Lab2-S0# ./ej1b
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
root@debian:/home/michael/Desktop/Lab2-S0# Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
Hello worlf!
```

- ¿Por qué hay tantos procesos en ambos programas cuando uno tiene cuatro llamadas `fork()` y el otro sólo tiene una?
  - Debido a que en los cuatro forks, cada uno llama a más forks dentro de él, de igual manera dentro del for hace que ese fork se corra 4 veces.

## Ejercicio 2(20puntos)

- ¿Cuál, en general, toma tiempos más largos?
  - En promedio el programa sin forks nos brinda un clock de 4000, mientras que en el programa con forks, cada uno toma en promedio 1200 y el padre unos 90.

```
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
3907.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
4096.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
3899.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
3930.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
4381.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
4295.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
4197.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
4121.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
4350.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
4095.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2a
4457.000000

root@debian:/home/michael/Desktop/Lab2-S0# ./ej2b
Hello from grand-grandchild: 1322.000000
Hello from grandchild: 1461.000000
Hello from child: 1447.000000
Hello from parent: 77.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2b
Hello from grand-grandchild: 1081.000000
Hello from grandchild: 1267.000000
Hello from child: 1138.000000
Hello from parent: 78.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2b
Hello from grand-grandchild: 1238.000000
Hello from grandchild: 1298.000000
Hello from child: 1268.000000
Hello from parent: 95.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2b
Hello from grand-grandchild: 1276.000000
Hello from grandchild: 1479.000000
Hello from child: 1319.000000
Hello from parent: 80.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2b
Hello from grand-grandchild: 1444.000000
Hello from grandchild: 1600.000000
Hello from child: 1486.000000
Hello from parent: 83.000000
root@debian:/home/michael/Desktop/Lab2-S0# ./ej2b
Hello from grand-grandchild: 1665.000000
Hello from grandchild: 1780.000000
Hello from child: 1665.000000
Hello from parent: 98.000000
```

- ¿Qué causa la diferencia de tiempo, o por qué se tarda más el que se tarda más?
  - La diferencia se da porque en el primer programa los fors tienen que esperar a que termine el anterior, mientras que en el segundo corren a la vez.

### Ejercicio 3(20puntos)

Average:	UID	PID	cswch/s	nvcswch/s	Command
Average:	0	1	123.55	0.00	systemd
Average:	0	2	0.05	0.00	kthreadd
Average:	0	10	45.16	0.00	rcu_sched
Average:	0	12	0.26	0.00	migration/0
Average:	0	16	0.26	0.00	migration/1
Average:	0	17	0.10	0.00	ksoftirqd/1
Average:	0	21	0.26	0.00	migration/2
Average:	0	22	0.05	0.00	ksoftirqd/2
Average:	0	26	0.26	0.00	migration/3
Average:	0	27	1.41	0.00	ksoftirqd/3
Average:	0	38	0.05	0.00	khugepaged
Average:	0	62	0.73	0.00	kworker/2:1-mm_percpu_wq
Average:	0	172	18.21	0.00	irq/18-vmwgfx
Average:	0	194	0.16	0.00	kworker/3:1H-kblockd
Average:	0	195	0.21	0.00	kworker/0:1H-kblockd
Average:	0	237	0.16	0.00	jbd2/sda1-8
Average:	0	455	23.08	0.00	systemd-logind
Average:	0	458	0.10	0.00	wpa_supplicant
Average:	1000	856	0.05	0.00	dbus-daemon
Average:	1000	890	158.40	2.88	gnome-shell
Average:	1000	908	0.31	0.00	Xwayland
Average:	1000	1030	0.05	0.00	gsd-color
Average:	1000	1110	0.05	0.00	gnome-software
Average:	1000	1671	85.71	0.10	gnome-terminal-
Average:	0	4282	0.78	0.00	kworker/0:4-mm_percpu_wq
Average:	0	7329	24.96	0.00	kworker/3:2-mm_percpu_wq
Average:	0	7395	0.42	0.00	kworker/u8:1-events_unbound
Average:	0	8091	157.61	0.00	kworker/u8:0-events_unbound
Average:	0	8124	19.15	0.00	kworker/1:0-events
Average:	0	8296	0.47	0.00	kworker/3:0-ata_sff
Average:	0	8318	0.99	231.03	pidstat
Average:	0	8319	96.76	0.00	kworker/u8:2-events_unbound

6.)

- ¿Qué tipo de cambios de contexto incrementa notablemente en cada caso, y por qué?
  - El número que aumenta es el de “gnome-terminal” que es justamente la terminal en la que estoy escribiendo en ese momento.



Average:	UID	PID	cswch/s	nvcswh/s	Command
Average:	0	1	0.58	0.00	systemd
Average:	0	9	0.25	0.00	ksoftirqd/0
Average:	0	10	80.58	0.00	rcu_sched
Average:	0	12	0.25	0.00	migration/0
Average:	0	16	0.25	0.00	migration/1
Average:	0	21	0.25	0.00	migration/2
Average:	0	26	0.25	0.00	migration/3
Average:	0	27	0.25	0.00	ksoftirqd/3
Average:	0	38	0.08	0.00	khugepaged
Average:	0	62	0.92	0.00	kworker/2:1-mm_percpu_wq
Average:	0	172	11.08	0.00	irq/18-vmwgfx
Average:	0	194	0.67	0.00	kworker/3:1H-kblockd
Average:	0	199	0.08	0.00	kworker/1:1H-kblockd
Average:	0	237	0.50	0.17	jbd2/sda1-8
Average:	0	450	0.08	0.00	alsactl
Average:	0	455	0.42	0.00	systemd-logind
Average:	0	458	0.08	0.00	wpa_supplicant
Average:	1000	890	42.75	3.00	gnome-shell
Average:	1000	908	0.25	0.00	Xwayland
Average:	1000	933	0.08	0.00	at-spi2-registr
Average:	1000	996	0.17	0.00	gsd-power
Average:	1000	1030	0.08	0.00	gsd-color
Average:	1000	1671	16751.00	8676.17	gnome-terminal-
Average:	0	4282	0.92	0.00	kworker/0:4-mm_percpu_wq
Average:	0	7329	2.50	0.00	kworker/3:2-mm_percpu_wq
Average:	0	7395	12874.25	0.50	kworker/u8:1-events_unbound
Average:	0	8124	12.00	0.00	kworker/1:0-events
Average:	1000	8315	0.25	0.00	bash
Average:	0	8319	73453.50	92.75	kworker/u8:2-events_unbound
Average:	0	8421	34670.83	1.08	kworker/u8:3-events_unbound
Average:	0	8425	0.50	0.00	kworker/3:0-ata_sff
Average:	0	8432	0.08	0.00	pidstat

Average:	0	10	41.96	0.00	rcu_sched
Average:	0	12	0.23	0.00	migration/0
Average:	0	16	0.23	0.00	migration/1
Average:	0	17	0.50	0.00	ksoftirqd/1
Average:	0	21	0.23	0.00	migration/2
Average:	0	22	0.05	0.00	ksoftirqd/2
Average:	0	26	0.23	0.00	migration/3
Average:	0	27	0.23	0.00	ksoftirqd/3
Average:	0	33	0.05	0.00	khungtaskd
Average:	0	38	0.09	0.00	khugepaged
Average:	0	62	0.82	0.00	kworker/2:1-mm_percpu_wq
Average:	0	172	6.99	0.00	irq/18-vmwgfx
Average:	0	194	0.77	0.00	kworker/3:1H-kblockd
Average:	0	199	0.09	0.00	kworker/1:1H-kblockd
Average:	0	200	0.09	0.00	kworker/2:1H-kblockd
Average:	0	237	0.54	0.09	jbd2/sda1-8
Average:	0	272	0.05	0.00	systemd-journal
Average:	0	296	0.05	0.00	systemd-udevd
Average:	101	337	0.05	0.00	systemd-timesyn
Average:	0	455	0.27	0.00	systemd-logind
Average:	104	457	0.05	0.00	dbus-daemon
Average:	0	458	0.09	0.00	wpa_supplicant
Average:	0	686	0.05	0.00	upowerd
Average:	1000	890	23.16	3.27	gnome-shell
Average:	1000	908	0.09	0.00	Xwayland
Average:	1000	933	0.05	0.00	at-spi2-registr
Average:	1000	996	0.05	0.00	gsd-power
Average:	1000	1110	0.05	0.00	gnome-software
Average:	1000	1671	4958.45	3550.68	gnome-terminal-
Average:	0	4282	0.82	0.00	kworker/0:4-events
Average:	1000	7071	0.05	0.00	gedit
Average:	0	7329	2.68	0.00	kworker/3:2-mm_percpu_wq
Average:	0	7395	19956.95	62.08	kworker/u8:1-events_unbound
Average:	0	8124	7.90	0.00	kworker/1:0-events
Average:	1000	8315	0.09	0.00	bash
Average:	0	8319	8437.69	7.40	kworker/u8:2-events_unbound
Average:	0	8421	10707.49	5.77	kworker/u8:3-events_unbound
Average:	0	8425	0.50	0.00	kworker/3:0-ata_sff
Average:	0	8480	0.05	0.00	pidstat
Average:	1000	8481	0.05	0.00	ej2b
Average:	1000	8482	902.63	10095.46	ej2b
Average:	1000	8483	749.82	13623.84	ej2b
Average:	1000	8484	962.62	4660.08	ej2b

Average:	UID	PID	cswch/s	nvcswh/s	Command
Average:	0	1	65.82	0.00	systemd
Average:	0	2	0.07	0.00	kthreadd
Average:	0	9	0.21	0.00	ksoftirqd/0
Average:	0	10	52.66	0.00	rcu_sched
Average:	0	12	0.25	0.00	migration/0
Average:	0	16	0.25	0.00	migration/1
Average:	0	17	0.84	0.00	ksoftirqd/1
Average:	0	21	0.28	0.00	migration/2
Average:	0	22	0.25	0.00	ksoftirqd/2
Average:	0	26	0.25	0.00	migration/3
Average:	0	27	0.56	0.00	ksoftirqd/3
Average:	0	33	0.04	0.00	khungtaskd
Average:	0	38	0.07	0.00	khugepaged
Average:	0	62	0.84	0.00	kworker/2:1-mm_percpu_wq
Average:	0	172	8.48	0.00	irq/18-vmwgfx
Average:	0	194	0.60	0.00	kworker/3:1H-kblockd
Average:	0	195	0.04	0.00	kworker/0:1H-kblockd
Average:	0	200	0.07	0.00	kworker/2:1H-kblockd
Average:	0	237	0.42	0.04	jbd2/sda1-8
Average:	0	455	0.11	0.00	systemd-logind
Average:	0	458	0.11	0.00	wpa_supplicant
Average:	0	461	0.04	0.00	cron
Average:	1000	890	89.33	231.85	gnome-shell
Average:	1000	908	0.18	0.00	Xwayland
Average:	1000	933	0.07	0.00	at-spi2-registr
Average:	1000	996	0.07	0.00	gsd-power
Average:	1000	1030	0.04	0.00	gsd-color
Average:	1000	1110	0.04	0.00	gnome-software
Average:	1000	1671	7235.52	2552.94	gnome-terminal-
Average:	0	4282	0.92	0.00	kworker/0:4-events
Average:	0	7329	2.04	0.00	kworker/3:2-mm_percpu_wq
Average:	0	7395	18608.52	7.39	kworker/u8:1-events_unbound
Average:	0	8124	9.50	0.00	kworker/1:0-events
Average:	1000	8315	0.07	0.00	bash
Average:	0	8421	20887.50	1.20	kworker/u8:3-events_unbound
Average:	0	8522	0.49	0.00	kworker/3:0-ata_sff
Average:	0	8526	0.99	67.30	pidstat
Average:	0	8527	4289.76	5.67	kworker/u8:0-events_unbound
Average:	0	8529	9217.56	165.58	kworker/u8:2-events_unbound



Average:	UID	PID	cswch/s	nvcswh/s	Command
Average:	0	1	29.40	0.00	systemd
Average:	0	9	0.70	0.00	ksoftirqd/0
Average:	0	10	61.84	0.05	rcu_sched
Average:	0	12	0.23	0.00	migration/0
Average:	0	16	0.23	0.00	migration/1
Average:	0	17	0.47	0.00	ksoftirqd/1
Average:	0	21	0.23	0.00	migration/2
Average:	0	26	0.23	0.00	migration/3
Average:	0	27	0.28	0.00	ksoftirqd/3
Average:	0	38	0.09	0.00	khugepaged
Average:	0	62	0.89	0.00	kworker/2:1-mm_percpu_wq
Average:	0	172	12.77	0.00	irq/18-vmwgfx
Average:	0	194	0.61	0.00	kworker/3:1H-kblockd
Average:	0	195	0.05	0.00	kworker/0:1H-kblockd
Average:	0	199	0.05	0.00	kworker/1:1H-kblockd
Average:	0	237	0.42	0.14	jbd2/sda1-8
Average:	0	455	0.14	0.00	systemd-logind
Average:	0	458	0.09	0.00	wpa_supplicant
Average:	1000	890	63.47	1.82	gnome-shell
Average:	1000	908	0.19	0.00	Xwayland
Average:	1000	933	0.09	0.00	at-spi2-registr
Average:	1000	1030	0.05	0.00	gsd-color
Average:	1000	1671	6684.06	3933.83	gnome-terminal-
Average:	0	4282	0.89	0.00	kworker/0:4-events
Average:	0	7329	2.05	0.00	kworker/3:2-events_freezable_power
<hr/>					
Average:	0	7395	15390.77	7.04	kworker/u8:1-events_unbound
Average:	0	8124	13.75	0.00	kworker/1:0-events
Average:	1000	8315	0.09	0.00	bash
Average:	0	8421	22472.18	1.54	kworker/u8:3-flush-8:0
Average:	0	8522	0.51	0.00	kworker/3:0-ata_sff
Average:	0	8529	20527.63	0.93	kworker/u8:2-events_unbound
Average:	0	8530	0.98	70.69	pidstat

12.)

- ¿Qué diferencia hay en el número y tipo de cambios de contexto de entre programas?
  - En el programa que utiliza forks se evidencian los diferentes procesos.
- ¿A qué puede atribuir los cambios de contexto voluntarios realizados por sus programas?
  - Las diferentes funciones que implementamos en el código.
- ¿A qué puede atribuir los cambios de contexto involuntarios realizados por sus programas?
  - Las llamadas al sistema que tiene que realizar el programa para correr nuestro código.
- ¿Por qué el reporte de cambios de contexto para su programa con `fork()` muestra cuatro procesos, uno de los cuales reporta cero cambios de contexto?
  - Debido a que es el proceso padre y dentro de él no había ningún ciclo for corriendo como en los otros procesos.



## Ejercicio 4(10puntos)

3.)

```
1 I      0 8735      2 0 80 0 - 0 - ? 00:00:00 kworker/3:0-event
0 R 1000 8736 8315 88 80 0 - 536 - pts/1 00:00:05 ej3a
1 Z 1000 8737 8736 0 80 0 - 0 - pts/1 00:00:00 ej3a <defunct>
4 R      0 8738 1819 0 80 0 - 2637 - pts/0 00:00:00 ps
```

- ¿Qué significa la Z y a qué se debe?

6.)

```
0 R 1000 8843 8315 99 80 0 - 536 - pts/1 00:00:03 ej3a
1 R 1000 8844 8843 16 80 0 - 569 - pts/1 00:00:00 ej3a
4 R      0 8846 1819 0 80 0 - 2637 - pts/0 00:00:00 ps
root@debian:/home/michael/Desktop/Lab2-S0# ps -ael
F S      UID      PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
```

- ¿Qué sucede en la ventana donde ejecutó su programa?
  - En lo personal no pude percibir ninguna diferencia en la ejecución al matar al proceso padre, sin embargo al matar al proceso hijo si paraban los números y salía la palabra “killed”.

7.)

```
1 I      0 8902      2 0 80 0 - 0 - ? 00:00:00 kworker/3:1-ata_
1 S 1000 8943 837 14 80 0 - 569 - pts/1 00:00:01 ej3a
4 R      0 8945 1819 0 80 0 - 2637 - pts/0 00:00:00 ps
4 S 1000 837 1 0 80 0 - 5348 do_ep0 ? 00:00:00 systemd
```

- ¿Quién es el padre del proceso que quedó huérfano?
  - Como se puede observar, ahora el proceso quedó con el proceso 837 como padre, que hace referencia a systemd.

## Ejercicio 5(40 puntos)

```
michael@debian:~/Desktop/Lab2-S0$ ./ipc
Enter the letter: m
Enter the number: 6
m
m
m
m
m
m
m
michael@debian:~/Desktop/Lab2-S0$
```

- ¿Qué diferencia hay entre realizar comunicación usando memoria compartida en lugar de usando un archivo de texto común y corriente?
  - De esta manera podemos acceder a la memoria y guardar información sólo de manera momentánea y requiere menos intervención del sistema operativo.
- ¿Por qué no se debe usar el file descriptor de la memoria compartida producido por otra instancia para realizar el mmap?
  - Debido a que produce problemas de acceso y sincronización.

- ¿Es posible enviar el output de un programa ejecutado con `exec` a otro proceso por medio de un pipe? Investigue y explique cómo funciona este mecanismo en la terminal (e.g., la ejecución de `ls | less`).
  - Si `exec` ejecuta el código como y permite enviar el output a otro pipe.
- ¿Cómo puede asegurarse de que ya se ha abierto un espacio de memoria compartida con un nombre determinado? Investigue y explique errno.
  - Si al intentar abrir el espacio de memoria nos devuelve Null significa que no se ha podido abrir el espacio.
- ¿Qué pasa si se ejecuta `shm_unlink` cuando hay procesos que todavía están usando la memoria compartida?
  - Se rompe la conexión que existe entre los procesos hacia la memoria compartida.
- ¿Cómo puede referirse al contenido de un espacio en memoria al que apunta un puntero? Observe que su programa deberá tener alguna forma de saber hasta dónde ha escrito su otra instancia en la memoria compartida para no escribir sobre ello.
  - Podemos hacer referencia a la memoria compartida utilizando llaves `[ ]` como si fuera una lista.
- Imagine que una ejecución de su programa sufre un error que termina la ejecución prematuramente, dejando el espacio de memoria compartida abierto y provocando que nuevas ejecuciones se queden esperando el file descriptor del espacio de memoria compartida. ¿Cómo puede liberar el espacio de memoria compartida “manualmente”?
  - Existe el comando `ipcrm` que se utiliza para eliminar recursos de la memoria compartida
- Observe que el programa que ejecute dos instancias de `ipcc` debe cuidar que una instancia no termine mucho antes que la otra para evitar que ambas instancias abran y cierren su propio espacio de memoria compartida. ¿Aproximadamente cuánto tiempo toma la realización de un `fork()`? Investigue y aplique `usleep`.

```
michael@debian:~/Desktop/Lab2-S0$ ./ipc
Enter the letter: m
Enter the number: 6
```

```
Tiempo: 0
```

- He corrido una prueba para ver el tiempo utilizando la herramienta de `clock` que utilizamos

en ejercicios anteriores, sin embargo ha salido un valor de 0, considero que quizás en este caso no se pueda utilizar de igual manera, de todas maneras se utilizó la función de usleep para esperar en el padre.