PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

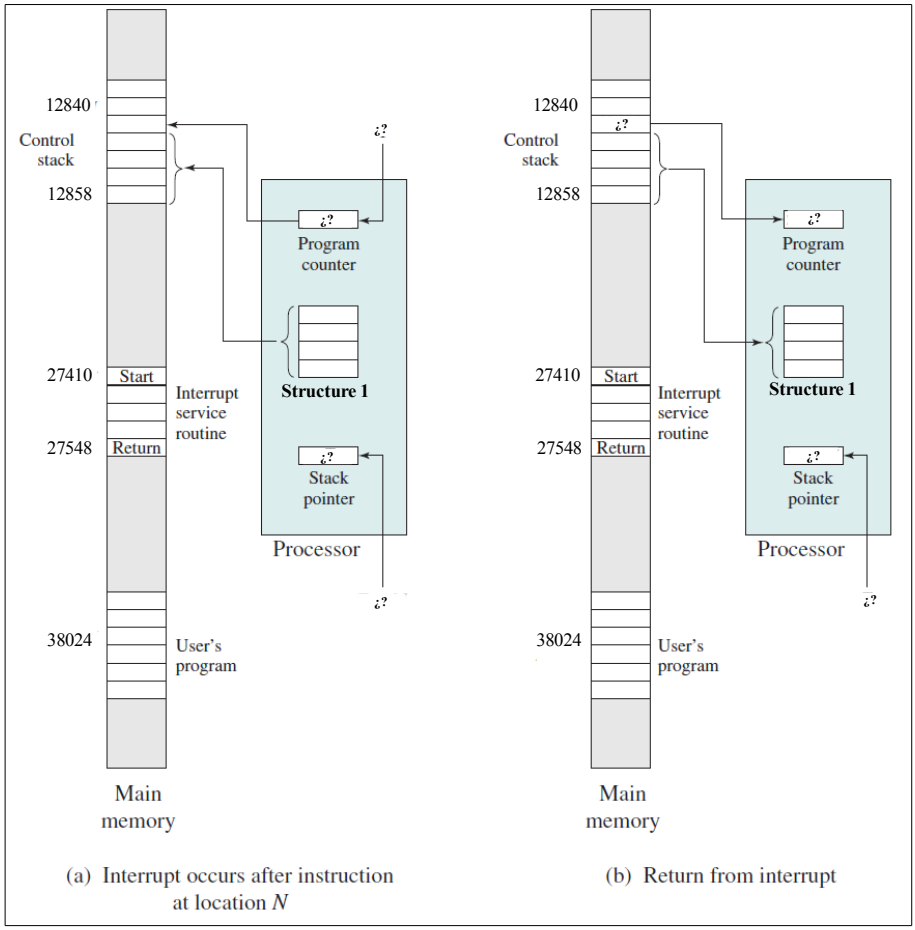
SISTEMAS OPERATIVOS

1ra práctica (tipo a)
(Primer semestre de 2017)

Horario 0781: prof. V. Khlebnikov

Duración: 1 h. 50 min.
Nota: No se puede usar ningún material de consulta.
La presentación, la ortografía y la gramática influirán en la calificación.
Puntaje total: 20 puntos

Pregunta 1 (5 puntos – 25 min.) En la siguiente figura se presentan los cambios en la memoria y en los registros en caso de una interrupción que llegó de un dispositivo mientras se ejecutaba la instrucción en la palabra 38024 de la memoria:



¿Qué es lo denotado en las figuras como "Structure 1"?

Indique los valores (denotados con "¿?") que están en y los que se cargan a los registros en ambas figuras.

Pregunta 2 (10 puntos – 50 min.)

Según el manual de `waitpid`, with the synopsis

```
pid_t waitpid(pid_t pid,
              int *status,
              int options);,
```

"This system call is used to wait for state changes in a child of the calling process, and obtain information about the child whose state has changed. A state change is considered to be: the child terminated; the child was stopped by a signal; or the child was resumed by a signal. In the case of a terminated child, performing a wait allows the system to release the resources associated with the child; if a wait is not performed, then the terminated child remains in a "zombie" state. … If a child has already changed state, then this call return immediately. Otherwise, it block until a child changes state. The value of `pid` can be `-1` meaning wait for any child process. … On success, `waitpid()`, returns the process ID of the child whose state has changed. On error, `-1` is returned."

Considere el código y la ejecución del siguiente programa:

```
$ cat -n 2017-1_pr1.c
     1  #include <sys/wait.h>
     2  #include <stdlib.h>
     3  #include <unistd.h>
     4
     5  int
     6  main(void)
     7  {
     8      pid_t pid;
     9      int status;
```

```
    10
    11      if (waitpid(-1,&status,0) != -1) exit(0);
    12      while (fork() + (pid=fork())) {
    13          if (!pid) {
    14              while (getppid() != 1);
    15              system("ps -l");
    16              exit(0);
    17          }
    18          execl("./2017-1_pr1","2017-1_pr1",NULL);
    19      }
    20      system("ps -l");
    21      exit(7);
    22  }
$ gcc 2017-1_pr1.c -o 2017-1_pr1
$ ./2017-1_pr1
```

¿Cuál es el árbol de los procesos creados durante la ejecución de este programa? ¿Cómo pasa la vida de cada proceso?

Pregunta 3 (2 puntos – 10 min.) Suppose that initially `x = 0` and we run a program with two threads that do the following:

Thread A	**Thread B**
`x += 1;`	`x += 2;`

What are the possible final values of `x`?

Pregunta 4 (3 puntos – 15 min.) The Too Much Milk problem models two roommates who share a refrigerator and who – as good roommates – make sure the refrigerator is always well stocked with milk. With such responsible roommates, the following scenario is possible:

	Roommate 1's actions	**Roommate 2's actions**
3:00	Look in fridge; out of milk.	
3:05	Leave for store.	
3:10	Arrive at store.	Look in fridge; out of milk.
3:15	Buy milk.	Leave for store.
3:20	Arrive home; put milk away.	Arrive at store.
3:25		Buy milk.
3:30		Arrive home; put milk away.
3:35		Oh no!

(a) (1 punto – 5 min.) A solution to the Too Much Milk problem must ensure both safety and liveness properties. What are these properties for this particular Too Much Milk problem?

Solution. The basic idea is for a roommates to leave a note on the fridge before going to the store. Each thread might run the following code:

```
if (milk == 0) {
    if (note == 0) {
        note = 1;
        milk++;
        note = 0;
    }
}
```

(b) (2 puntos – 10 min.) Unfortunately, this implementation can violate one of the solution's established properties. Which one and how?

La práctica ha sido preparada por VK
con LibreOffice Writer en Linux Mint 18.1 Serena.

Profesores del curso: (0781) V. Khlebnikov

Pando, 28 de abril de 2017