Signal Quizzes

Exercise

Given the following segments of pseudo-code, which are executed concurrently by processes P1 and P2 with PID pid_P1 and pid_P2, respectively, indicate which of the following statements are TRUE.

```
P1
while (1) {
       ...
       kill (pid_P2, SIGUSR1);
       pause ();
       A();
}

P2
while (1) {
       pause ();
       B();
       ...
       kill (pid_P1, SIGUSR2);
}
```

Choose one or more options:

1. □ It is possible to execute function A() more than one time (without the execution of any B() function in between)
2. □ Function A() can be executed before function B()
3. □ They are subject to deadlocks
4. □ They are subject to starvation
5. □ Function B() is certainly executed at least one time
6. Function B() is always executed before function A()

Correct Answers:
3, 4

Exercise

Describe the characteristics of the signals to send asynchronous information between processes. Implement the following program:

1. A program generates two processes P1 and P2 and it awaits for receiving signals from them. Every time it receives a message from P1 in displays on standard output the message "Signal received from P1". Analogously, every time it receives a message from process P2 it displays the message "Signal received from P2". If it receives 3 signals from the same process, it terminates processes P1 and P2 using the shell command "kill" and it terminates.

2. Process P1 and P2 run through an infinite cycle. Within the cycle, they await for a random time and then send a signal to the parent. P1 sends signal SIGUSR1; P2 sends signals SIGUSR2.

Answer:

```
#include<signal.h>
#include<sys/types.h>
#include<sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
```

```
int last_sig = -1;
int last_last_sig = -1;
int finish = 0;

void sign_handler(int sig){
      if (sig==SIGUSR1)
            printf("Signal received from P1\n");
      else if (sig==SIGUSR2)
            printf("Signal received from P2\n");

      /* It receives 3 consecutive signals from the same process */
      if (sig == last_sig && last_sig == last_last_sig) {
            finish = 1;
      } else {
            last_last_sig = last_sig;
            last_sig = sig;
      }
}

int main() {
      char cmd[100];
      pid_t pid1, pid2;
      if ( (signal(SIGUSR1, sign_handler) == SIG_ERR) || (signal(SIGUSR2,
      sign_handler ) == SIG_ERR) ) {
            printf("Error initializing signal handler");
            exit(-1);
      }
      pid1 = fork();
      if (!pid1) {
            /* P1 */
            while (1) {
                  sleep( rand()%2 );
                  kill(getppid(), SIGUSR1);
            }
      } else {
            pid2 = fork();
            if (!pid2) {
                  /* P2 */
                  while (1) {
                        sleep( rand()%3 );
                        kill(getppid(), SIGUSR2);
                  }
            }
      }
      /* Parent */
      while (1) {
            pause();
            if (finish) {
                  /* Kill P1 with a shell command */
                  sprintf(cmd, "kill -9 %d", pid1);
                  system(cmd);
                  /* Kill P2 with a shell command */
                  sprintf(cmd, "kill -9 %d", pid2);
                  system(cmd);
                  exit(0);
            }
      }
}
```

Exercise

Refer to the system call kill and to the kill shell command.
Indicate which of the following statements are correct. Note that wrong answers imply a penalty in the final score.

Choose one or more options:

1. □ The sent signal can be masked only in the case of the system call
2. □ Both the shell command and the system call can kill a process
3. □ The system call kills a process, the shell command does not
4. □ Both send a signal to a process
5. □ The shell command kills a process, the system call does not

Correct Answers:
2, 4

Exercise

Suppose to execute the following program with the value 4 passed on the command line. Report the output generated by the program. Suppose to run the program in an operating system where waiting of 1 second is sufficient to allow all other tasks that are running in that specific moment of time to terminate. Please, report the response on a single line, indicating the various messages and output values separated by a single space. Do not insert any other character into the answer. Format errors will be treated in the same way as other errors. This is an example of a correct answer: a 4 b++ etc.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(){
      int i;
      for (i=0; i<4 && !fork(); i++){
            if (fork()) {
                  sleep (1);
                  system ("echo i+");
            }
            execlp ("echo", "system", "i++", NULL);
      }
}
```

Correct Answer:
i++ i+ i++

Exercise

Refer to the use of signals in a UNIX/Linux environment.
Indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

Choose one or more options:

1. □ By using the system call signal() you can decide to ignore ANY type of signal
2. □ Inside a signal handler, only reentrant functions should be used
3. □ The reception of a signal by a process ALWAYS causes the termination of the process

4. □ The use of the pair of functions kill() and pause() or of the semaphore primitives sem_post() and sem_wait() are EQUIVALENT from the point of view of process synchronization
5. □ The reception of some types of signal (for instance SIGKILL) cannot be ignored
6. □ The execution of a signal handler after the reception of a signal by a process can lead to race conditions

Correct Answers
2, 5, 6

Exercise

Suppose a process executes the following instruction:

```
signal(SIGCHLD, SIG_IGN);
```

Indicate which ones among the following observations are correct (possibly more than one). Note that incorrect answers may imply a penalty on the final score.

Choose one or more options:

1. □ The process will handled signals of type SIGCHLD with the default function SIG_IGN (which is defined as a macro in signal.h).
2. □ The process will not have to execute a wait or a waitpid.
3. □ The process will behave in a standard way when a SIGCHLD is received.
4. □ The process will ignore signals of type SIGCHLD.
5. □ If the process executes a wait, it will return an error code.
6. □ The child process can become zombie.

Correct Answers
1, 2, 4, 5

Exercise

Indicate which of the following statements related to signals are correct. Note that incorrect answers imply a penalty in the final score.

Choose one or more options:

1. □ If a process receives a signal and then it executes a pause(), it will remain blocked on it (as long as it does not receive another signal).
2. □ After the reception of a signal the process has three possibilities: accept the default behavior, ignore it (not for all signals), or catch the signal by defining a signal handler.
3. □ Signal can be transferred only between processes with the same parent.
4. □ The use of the kill() always leads to the termination of the process to which it is addressed.
5. □ The system call signal() can be used to send a signal.

Correct Answers
1, 2

Exercise

Given three processes P1, P2 and P3 whose code is reported in the following and whose pids are pid_P1, pid_P2, and pid_P3, respectively. Indicate which of the following outputs are correct. Note that incorrect answers imply a penalty in the final score.

```
P1
```

```
kill(pid_P2, SIG…);
pause();
printf("A");
P2
other_code();
pause();
printf("B");
kill(pid_P1, SIG…);
kill(pid_P3, SIG…);
P3
other_code();
pause();
printf("C");
```

Choose one or more options:

1. □ No output.
2. □ B
3. □ BC
4. □ BA
5. □ BCA
6. □ BAC

Correct Answers
1, 2, 3, 4, 5, 6