Operating System 2021
Quiz #3

- What is the content of the shell variable $? after running the following program?

```c
#include <unistd.h>

int main(void)
{
  execlp("true", "false", NULL);
  return 42;
}
```

  - 0

- How many times does the following C program prints yes?

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

int main(void)
{
  fork(); fork(); fork();
  puts("yes");
  return 0;
}
```

  - Eight times

- What is the output produced by the following program?

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

int main(void)
{
  int i = 1;

  if (! fork() || fork()) {
     fork();
  }
  printf("%d ", i++);
  fflush(stdout);
  return 0;
}
```

  - 1 1 1 1 1

- What is the content of the shell variable $? after running the following program?

```c
#include <unistd.h>
#include <sys/wait.h>

int main(void)
{
  pid_t pid;
  int status = 0;

  pid = fork();
```

```c
    if (pid == -1) {
        status = 2;
        return status;
    }
    if (pid == 0) {
        status = 3;
        execlp("false", "false", NULL);
        return 1;
    }
    (void) waitpid(pid, &status, 0);
    return WEXITSTATUS(status);
}
```

- 1

- What is the content of the shell variable $? after running the following program?

```c
#include <unistd.h>
#include <sys/wait.h>

int main(void)
{
  pid_t pid;
  int status = 0;

  pid = fork();
    if (pid == -1) {
        return 1;
    }
    if (pid == 0) {
        return 2;
    }
    (void) waitpid(pid, &status, WNOHANG);
    return WEXITSTATUS(status);
}
```

- 0 or 1 or 2

- If a thread opens a file for reading with read privileges, then other threads in the same process can also read from that file.
- the following memory segments are not shared between threads of the same process
    - stack segment
- process control blocks
    - On many operating systems, it is impossible to print a consistent snapshot of the list of running processes while the operating system is running., The content of processor registers can be saved in the process control block if a process got interrupted by the operating system kernel., Processes have a unique numeric identifier, which is stored in the process control block and commonly used to refer to a specific process.
- process states
    - The state of a process is stored in its process control block., Processes stay in the terminated state until all resources have been released and the exit status code has been delivered to the parent process.
- A process has at least one control flow and it owns an internal and external state., The operating system kernel isolates processes: data managed by a process is protected

from access by other processes., Every process maintains its own heap and stack memory segments.

- system calls initialize the stack of the calling process
    - exec()