laSalle
UNIVERSITAT RAMON LLULL

# Operating Systems 2023-2024

**Session 6:** Pipes & SM 02/11/23

## Objective

In this session you will begin to practice complex techniques specific to operating systems. This session will focus on shared memory usage as well as other tools seen in previous sessions like pipes.

## Motivation

More specifically, with this session, the student will exercise:

- Creation of shared memory (shm*get, shmat*).
- Destruction of shared memory (*shmdt*, shm*ctl*).
- Access to the shared memory.
- Creation of pipes (*pipe*)

## Previous documentation

To complete this session, it is recommended to read of the following references:


SALVADOR, J. (2014). *Programació en UNIX per a pràctiques de Sistemes Operatius*.

STEVENS, W. R. & RAGO S.A. (2008). Advanced Programming in the UNIX Environment, 2nd edition.

TANENBAUM, A. S. (2009). Modern Operating Systems, 3rd Edition.

# Pool Control

Every summer is getting warmer; therefore, people want more pools, the water of the pools require a strict control to prevent algae. Traditionally this kind of controls have been done in a manual way, but it is as tedious job outside at the sun. We have been asked to automate this control system.

The system processes up to 100 liters **every cycle**, the water has already been analyzed and we only need to make the necessary processes to keep it clean. We are provided a file with the results of the analysis, the 1st line is the total number of litters to treat, the following lines are the analysis results for each cycle. The analysis provides 5 values separated by an underscore and in the following order:

1. Number of big particles/$m^3$.
2. Number of small particles/$m^3$.
3. Alkalinity. Ideal value is 100, correct values are between 80 and 120.
4. pH. Ideal value is 7.4, correct values are between 7.2 and 7.6.
5. Chlorine. Ideal value is 2, correct values are between 1 and 3.

Example file:

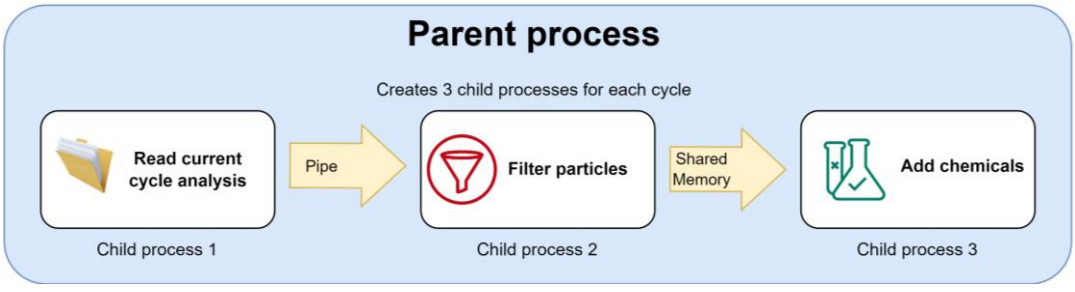
```
225
30_500_95.7_6.4_0
26.58_456_63_8.4_2
45_625.3_120.13_7.4_1

```

analysis.txt

The system consists of 3 different phases, each of them must be a different process:

1. Read the analysis file and send the data for the coming cycle to the second phase via a pipe and in binary format.
2. Pumping the water and removing all kind of particles. The process should remove 10 big particles every second and 140 small particles every second[1]. The analysis data must be sent to the next process via shared memory.
3. Add the necessary chemicals to bring water to ideal Alkalinity, pH and chlorine values.




[1] Running times cannot be rounded down, for 15 big particles and 182 small particles the system must be running at least for 1.5 and 1.3 seconds respectively.

# Execution example

**NOTE:** analysis.txt contains the example shown previously in this statement.

marc.valsells@montserrat:~> ./S6 analysis.txt

```
----- Starting to process a cycle -----

Starting phase 1...

Starting phase 2...
Removing 30.000000 big particles in 3 seconds...
Removing 500.000000 small particles in 4 seconds...

Starting phase 3...
Current cycle alkalinity is 95.699997. Value OK, doing nothing.
Current cycle ph is 6.400000. Adding 1.000000 ph...
Current cycle chlorine is 0.000000. Adding 2.000000 chlorine...

----- Starting to process a cycle -----

Starting phase 1...

Starting phase 2...
Removing 26.580000 big particles in 3 seconds...
Removing 456.000000 small particles in 4 seconds...

Starting phase 3...
Current cycle alkalinity is 63.000000. Adding 37.000000 alkalinity...
Current cycle ph is 8.400000. Removing 1.000000 ph...
Current cycle chlorine is 2.000000. Value OK, doing nothing.

----- Starting to process a cycle -----

Starting phase 1...

Starting phase 2...
Removing 45.000000 big particles in 5 seconds...
Removing 625.299988 small particles in 5 seconds...

Starting phase 3...
Current cycle alkalinity is 120.129997. Removing 20.129997 alkalinity...
Current cycle ph is 7.400000. Value OK, doing nothing.
Current cycle chlorine is 1.000000. Value OK, doing nothing.
```

laSalle
UNIVERSITAT RAMON LLULL

# Operating Systems 2023-2024

**Session 6:** Pipes & SM 02/11/23

## Considerations

- All child processes must be created using forks.
- Communication between processes must be done with Pipes or Shared Memory. All their resources must be freed.
- It's very important to free the memory properly in this session, as well as terminating all processes.
- Use **ps -u** to view all your active processes. If you wish to terminate all processes with the same name, all at once (in case you forgot to terminate al the child processes during the execution), we recommend you use the command **killall <process name>** *(if you executed the processes with ./ at the start, don't include them on the killall command)*
- It must be ensured that at the end of the execution all the processes have ended correctly without leaving zombie or orphaned processes.
- The execution output must be identical as the example provided.
- The use of "**system**" or "**popen**" or similar functions is not allowed.
- All input and output must be performed using file descriptors. The use of **printf**, **scanf**, **FILE***, **getchar**, or similar is not allowed.
- It must be compiled using the flags –Wall and –Wextra. If using the math.h library the flag -lm might need to be added.
- Any session that contains warnings will be discarded.
- At the start of the .c you must include a comment with your logins, names, and surnames.
- For submitting the session, you must hand over a file "S6.c" and deliver it through eStudy platform.