

# Hello Dolly

---

ID1206 OPERATING SYSTEMS

ISABEL GHOURCHIAN

2017



# Intro - processes

---

- A process is a running program, an execution of instructions and a set of resources
- Can be represented by a unique number called process id (pid)

# fork()

---

- Function that creates a new process
- Creates a copy (child process) of the current process (parent process)
- Returns twice
  - child's pid to the parent
  - 0 to the child
- Child process has its own address space

# Example – fork()

---

```
int pid = fork();  
if (pid == 0) {  
    printf("I'm the child");  
}  
else {  
    printf("I'm the parent ");  
}
```

# Zombie

---

- Child process that has finished executing but still exists in the process table
- Happens if parent does not call wait()

```
else {                                //parent process
    printf("I'm the parent");
    wait(NULL);                        //wait for child process to terminate, prevent zombie
}
```

# Orphan

---

- Child process that is still running but the parent process has finished executing or terminated
- Will be "adopted" by another process, get a new parent process
- In UNIX the new parent process is the *init* or *systemd* process
  - parent of all processes
  - process id 1

# Daemon

---

- Process that runs automatically in the background instead of under direct control of a user
- Performs operations at predefined times or in response to events
- Runs most of the tasks in a system

# exec()

---

- Runs an executable file in the context of an existing process, replacing the previous execution context
- `execlp("ls", "ls", NULL);`
  - check if 'ls' exists
  - run it with the rest of the arguments
- exec functions do not return when successful



# exec() family of functions

---

- execl, execlp, execl, execv, execvp, execvpe
  - e: pass an array of pointers to environment variables
  - l: arguments are passed individually to the function
  - v: arguments are passed as an array of strings
  - p: uses PATH environment variable to find the file that is to be executed

# Pipes

---

- Sends the output of one program to another programs input
- Denoted by symbol '|'
- Piping in the shell: combine several commands
  - Ex: `cat countries.txt | grep a | sort`
  - Displays all countries that start with an 'a' and sorts them in alphabetic order