

# Proyecto 1

Roberto Artigues, Emilio Meza, Nicolás Soto

8 de Octubre de 2023

## 1. Introducción

Desarrollo de un intérprete de comandos simple en Linux (shell). La shell a implementar será similar a las disponibles actualmente en Linux.

## 2. Parte 1

En el main del código se pueden visualizar todos los procesos solicitados para este proyecto, cada uno comentado para que sea más fácil su entendimiento.

## 3. Parte 2

Se creó la función a continuación, con sus comentarios en el código explicando los pasos que sigue.

```
void start_daemon(int t, int p) {
    pid_t pid = fork(); // Se crea un proceso hijo
    if (pid == 0) {
        umask(0); // Cambiar la máscara de modo de archivo, para que los archivos creados por el demonio tengan todos los permisos
        setsid(); // Nueva sesión
        close(STDIN_FILENO);
        close(STDOUT_FILENO);
        close(STDERR_FILENO);

        openlog("SusDaemon", LOG_PID, LOG_DAEMON); // Iniciar el log
        syslog(LOG_NOTICE, "El diablillo se ha creado"); // Mensaje de inicio

        for (int i = 0; i < p / t; ++i) {
            syslog(LOG_NOTICE, "Diablillo corriendo por la %d vez", i + 1);
            std::ifstream file("/proc/stat"); // Obtenemos la info desde /proc/stat porque /proc/cpuinfo no tiene la info de procesos (Al menos en AWS Ubuntu)
            // std::ifstream file("/proc/cpuinfo"); // Descomentar esta línea para probar con /proc/cpuinfo
            std::string line;
            int processes = 0, procs_running = 0, procs_blocked = 0;

            if (file.is_open()) {
                while (std::getline(file, line)) {
                    if (line.find("processes") != std::string::npos) {
                        std::istringstream iss(line);
                        std::string key;
                        iss >> key >> processes;
                    }

                    if (line.find("procs_running") != std::string::npos) {
                        std::istringstream iss(line);
                        std::string key;
                        iss >> key >> procs_running;
                    }

                    if (line.find("procs_blocked") != std::string::npos) {
                        std::istringstream iss(line);
                        std::string key;
                        iss >> key >> procs_blocked;
                    }
                }
                file.close();
            }

            syslog(LOG_NOTICE, "Procesos: %d, Corriendo: %d, Bloqueados: %d", processes, procs_running, procs_blocked);
            sleep(t);
        }
        syslog(LOG_NOTICE, "Diablillo ha terminado");
        closelog(); // Cerrar el log
        exit(0);
    }
}
```

Figura 1: Función Demonio

```
RobLand SusDaemon[16761]: El diablillo se ha creado  
RobLand SusDaemon[16761]: Diablillo corriendo por la 1 vez  
RobLand SusDaemon[16761]: Procesos: 17114, Corriendo: 3, Bloqueados: 0  
RobLand SusDaemon[16761]: Diablillito ha terminado
```

Figura 2: Demonio haciendo sus procesos