

ITBA - SISTEMAS OPERATIVOS

---

# Trabajo Práctico 1

## Inter Process Communication

---

*Alumnos:*

De Simone, Franco

61100

Dizenhaus, Manuel

61101

Cornídez, Milagros

61432

# 1 Introducción

Para este primer trabajo practico, se debió implementar un sistema constituido por 3 procesos principales: master, slave, y view. Estos 3 procesos debían comunicarse a través de los diferentes mecanismos de Inter Process Communication (IPC) desarrollados en clase, tales como pipes, shared memory, entre otros.

## 2 Aclaración

Al ser la segunda vez que realizamos este trabajo, el informe será breve y se enfocará en las correcciones realizadas al trabajo realizado el cuatrimestre anterior (cuyo repostorio se puede encontrar en [este vínculo](#)<sup>1</sup>),. Tomando como referencia las correcciones mencionadas, realizamos diferentes modificaciones siguiendo los lineamientos indicados por la cátedra.

### 2.1 Funcionalidad

Dado que en la entrega pasada prácticamente no hubieron errores de funcionamiento, decidimos mantener la estructura del proyecto tal y como estaba. Existía un problema cuando se enviaban diversos archivos a la vez, pero no se pudo encontrar el problema que lo generaba, por lo que preferimos mantener el funcionamiento ya correcto.

### 2.2 Calidad de código y limpieza de recursos del sistema

Tras la entrega el cuatrimestre anterior, donde la calidad de código nos costo todos los puntos, decidimos tomarnos el tiempo de leer todas las correcciones indicadas y modificar todo lo necesario para que no ocurra lo mismo. Pasando en orden de correcciones, lo primero que realizamos fue modificar todas las variables para que pasen a estar en *Camel Case*. Luego, se cambiaron variables con nombres poco descriptivos tales como *p*, *c* a variables con nombres que denoten su utilidad, y también se consideró la cuestión de la mezcla entre idiomas de variables.

Otro de nuestros errores principales fue haber corrido el código con cpp y valgrind de manera muy tardía en la entrega anterior, por lo que esta vez nos tomamos el tiempo de realizar el análisis de cada uno, y dejamos un archivo llamado "cpp-valgrind-PVS.txt" donde se especifican los errores arrojados por tanto cpp y PVS studio (análisis estático) como por Valgrind (análisis dinámico).

---

<sup>1</sup>Vínculo a repositorio de cuatrimestre anterior: [https://github.com/ManuelDizen/TP1\\_S0](https://github.com/ManuelDizen/TP1_S0)

## 2.3 Consideraciones generales

Al final del trabajo también se marcaron cosas a considerar, que tuvimos en cuenta esta vez:

- Se incluyó un *.gitignore*
- Se modularizó en carpetas, aunque en este caso el *scope* del trabajo es bastante limitado. Se dejaron los *.h* en la carpeta respectiva.
- Se incluyó en el makefile la instalación del *minisat*

**Nota:** Se aclaró en el makefile, pero es importante notar que la instalación del mismo se consideró para el entorno *docker*. Lógicamente, si uno desea instalar un paquete en su máquina local, se ve obligado a realizarlo anteponiendo *\$ sudo*.