



Universidad  
Nacional  
de Córdoba

# Cátedra de Sistemas Operativos II

## Trabajo Práctico N° I

---

Soriano, Juan

11 de abril de 2019

## Índice

<b>Introducción</b>	<b>3</b>
Propósito	3
Definiciones, Acrónimos y Abreviaturas	3
Referencias	3
<b>Descripción General</b>	<b>4</b>
Perspectiva del Producto	4
Funciones del Producto	5
Características de los Usuarios	5
Restricciones	5
Suposiciones y Dependencias	5
Requisitos Futuros (preguntar)	5
<b>Requisitos Específicos</b>	<b>6</b>
Interfaces Externas	6
Funciones	6
Requisitos de Rendimiento	7
Restricciones de Diseño	7
Atributos del Sistema	7
<b>Implementación y Resultados</b>	<b>8</b>
<b>Conclusiones</b>	<b>8</b>

## Introducción

En esta sección se encuentra la especificación de los requerimientos de la aplicación Sockets correspondiente al trabajo N° 1 de la Sistemas Operativos II.

La definición de un requerimiento se establece como las características que debe tener el sistema o las restricciones que debe satisfacer para la aceptación del cliente. Por esta razón, la especificación de los requerimientos y de las restricciones en este documento, está escrita de manera tal que el cliente las pueda comprender.

## Propósito

El propósito del Trabajo Práctico nº 1 de Sistemas Operativos II es el de diseñar e implementar un software que utilice la API de sockets del Sistema Operativo aplicando los conocimientos adquiridos durante la materia, así también como en Sistemas Operativos I e Ingeniería de Software.

## Definiciones, Acrónimos y Abreviaturas

Login: Ingreso.

API : Interfaz de Programación de Aplicaciones

## Referencias

1. Linux man pages: <https://linux.die.net/man/>
2. GNU, "5 Making The Best Use of C",  
[https://www.gnu.org/prep/standards/html\\_node/Writing-C.html](https://www.gnu.org/prep/standards/html_node/Writing-C.html)
3. W. Richard Stevens, Stephen A. Rago, Advanced Programming in the UNIX Environment, 3rd Edition, Addison-Wesley

## Descripción General

En la presente sección se explicarán los distintos aspectos del trabajo. Las funciones que ofrece el producto, la perspectiva del mismo, las características de los usuarios, requisitos para futuras iteraciones, entre otras cosas.

### Perspectiva del Producto

Según Wikipedia: “Un Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

El término socket es también usado como el nombre de una interfaz de programación de aplicaciones (API) para la familia de protocolos de Internet TCP/IP, provista usualmente por el sistema operativo.

Los sockets de Internet constituyen el mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos apropiados. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.”

La idea del producto es emular el comportamiento de un sistema de satélites que se conectan a una base terrena para intercambiar información. Un usuario ingresa al sistema hospedado en la base terrena y una vez autenticado, envía comandos a los satélites conectados para realizar acciones.

El presente producto se presenta en forma de dos programas en lenguaje C ejecutables. Uno que se ejecutará en el servidor y otro en el cliente.

Servidor: Se implementan las funcionalidades necesarias para la autenticación del cliente que realiza la conexión, manejo de múltiples clientes en simultáneo y transferencia de archivos.

Cliente: Se implementan las funcionalidades necesarias para establecer una conexión de forma segura (TCP) para el intercambio de mensajes y una conexión no orientada a la conexión (UDP) para el envío de información de telemetría

## Funciones del Producto

Utilizando Sockets, se podrán establecer múltiples conexiones TCP entre múltiples clientes y un único servidor. Previa a la conexión entre los clientes y el servidor, se lleva a cabo una etapa de autenticación en la cual el usuario se identifica e intenta loguearse al sistema. Si no es aceptado por el servidor, se le pedirá que vuelva a autenticarse hasta un máximo de tres intentos.

Una vez llevada a cabo la etapa del login, el usuario recibe el prompt del programa que indica que la base terrena se encuentra recibiendo comandos. Una vez recibido el prompt, el usuario puede realizar una de tres acciones: recibir telemetría del satélite por medio de UDP, recibir un escaneo de la tierra por protocolo TCP o enviar un update de firmware al satélite para que éste se actualice y se reinicie con la versión actualizada.

## Características de los Usuarios

El producto se destina a usuarios con conocimientos básicos del manejo de sistemas operativos Linux por consola.

## Restricciones

Las restricciones son muy bajas. Se requiere de una computadora con requerimientos de hardware mínimos capaz de correr el programa del servidor y de la misma forma, una computadora con requerimientos de hardware mínimos que ejecute el programa del cliente. Debe haber conectividad de red entre ambas partes. En este caso, el programa cliente se ejecuta en una placa raspberry pi pero puede realizarse en cualquier placa que tenga una MMU.

## Suposiciones y Dependencias

La performance del producto se encuentra atada a la calidad de la conexión entre cliente y servidor, por lo que si la calidad de la conexión es mala, los resultados obtenidos no serán de buen calibre.

## Requisitos Futuros

Queda un gran abanico de posibles funcionalidades para ser agregadas, a continuación se listan algunas de ellas:

- Mejorar el sistema de autenticación con el fin de hacerlo más robusto y seguro.
- Solucionar problemas de concurrencia que ocurren cuando múltiples usuarios se encuentran conectados en simultáneo.

- Control de integridad de la telemetría recibida por medio de una función de hash como puede ser SHA1.

## Requisitos Específicos

En la presente sección se explicará con mayor lujo de detalle las entradas y salidas del sistema como así también una descripción de las funciones y requerimientos funcionales del producto.

### Interfaces Externas

A continuación se ofrece una breve descripción de las interfaces implementadas.

#### **Interfaz de usuario**

La interfaz que se ofrece al usuario es un Prompt en una consola. Esto se hizo para simplicidad del uso del programa por parte del usuario y para reducir el volumen de errores.

#### **Sockets**

Para la comunicación se utilizaron tanto Sockets TCP como UDP.

### Funciones

En esta sección se listan los requerimientos funcionales de la aplicación para la versión 1 del release del producto con fecha 11 de Abril de 2019:

1. Conexión local al programa del servidor para autenticarse.
2. Manejo de multiusuario por parte del servidor, proveyendo la escucha a múltiples conexiones de distintos clientes en simultáneo.
3. Autenticación con usuario y contraseña del usuario que se realiza del lado del servidor.
4. Verificación de la validez del usuario y la contraseña del cliente del lado del servidor. Que resulte en una notificación al cliente con la leyenda “Usuario y/o contraseña incorrectos”
5. Prompt del lado del servidor que le indique al usuario que hay conexión con los satélites.
6. Envío de comandos de forma remota desde el servidor hacia el cliente o clientes.
7. Intercambio de información entre el cliente o clientes y el servidor.
8. Obtención de telemetría por medio de sockets UDP. El servidor obtiene el ID del satélite, el uptime del mismo, su memoria libre y su versión de firmware.
9. Recepción de imagen de escaneo por parte del servidor desde el cliente. El satélite envía una imagen por TCP de la tierra hacia la base terrena.

10. Update del firmware de los satélites. La base terrena envía un update de una nueva versión del programa a los satélites conectados. Luego éstos se reinician con la nueva versión del firmware.

## Requisitos de Rendimiento

En esta sección se listan los requerimientos funcionales de la aplicación para la versión 1 del release del producto con fecha 11 de Abril de 2019:

1. Uso de memoria mínimo y apropiado para que el programa pueda ser corrido en una placa de desarrollo con MMU, como puede ser una placa Raspberry Pi.
2. Tiempo de respuesta del prompt en el servidor por el usuario de menos de 200 ms.

## Restricciones de Diseño

En esta sección se listan las restricciones de diseño de la aplicación para la versión 1 del release del producto con fecha 11 de Abril de 2019:

1. Tanto el código del cliente como el del servidor serán desarrollados en lenguaje C.
2. Se utilizará el puerto 2019 como puerto para la conexión TCP, puerto en el cual se encontrará en todo momento escuchando el servidor una vez realizada la autenticación.
3. El servidor estará montado en una placa de desarrollo que soporte sistemas operativos del tipo GNU/Linux.
4. Sintaxis de código del tipo GNU o Linux Kernel.

## Atributos del Sistema

En esta sección se listan los atributos del sistema de la aplicación para la versión 1 del release del producto con fecha 11 de Abril de 2019:

1. Sistema portable, es decir, deberá ser capaz de correr en cualquier sistema operativo.
2. Sistema compilado con GCC y flags -Werror -Wall -pedantic
3. Se correrá la herramienta de análisis de código Cppcheck.

## Implementación y Resultados

Inicialmente se implementó el sistema utilizando sockets del tipo UNIX en una máquina local. Luego se llevó a la implementación utilizando sockets de internet del tipo TCP y UDP. Para el caso del envío de telemetría del satélite a la base terrena, se utilizó socket UDP. En los otros casos, con los comandos “start scanning” y “update firmware.bin” se utilizaron sockets TCP. Además, en el caso del envío de la imagen, se compararon los tiempos de recepción del lado de la base terrena y se cambiaron varios factores. Por un lado se modificó el tamaño del buffer de envío y recepción a nivel aplicación, y por otro lado, mediante `setsockopt()`, se modificó el tamaño del buffer de TCP que se puede ver en `/proc/sys/net/ipv4/tcp_mem`. En el caso de la Raspberry Pi, el tamaño máximo era más pequeño. Por más que se modificaron estos parámetros, no se notó diferencia alguna en la velocidad de transferencia de la imagen.

## Conclusiones

A modo de conclusión se puede decir que si bien al trabajo le quedaron varios aspectos por mejorar, se logró el cometido que es el de utilizar los conceptos aprendidos en diversas materias de la carrera. Se aplicaron los conocimientos de comunicación por sockets, comunicación entre procesos y muchos otros.

Al implementarse el trabajo en lenguaje C, se presentan diversos problemas cuya resolución no es trivial. Sin embargo, se pudieron sortear los obstáculos y se logró un resultado satisfactorio.