



Universidad
Nacional
de Córdoba

Cátedra de Sistemas Operativos II

Trabajo Práctico N° I

Salvatierra, Andrés
12 de abril de 2019

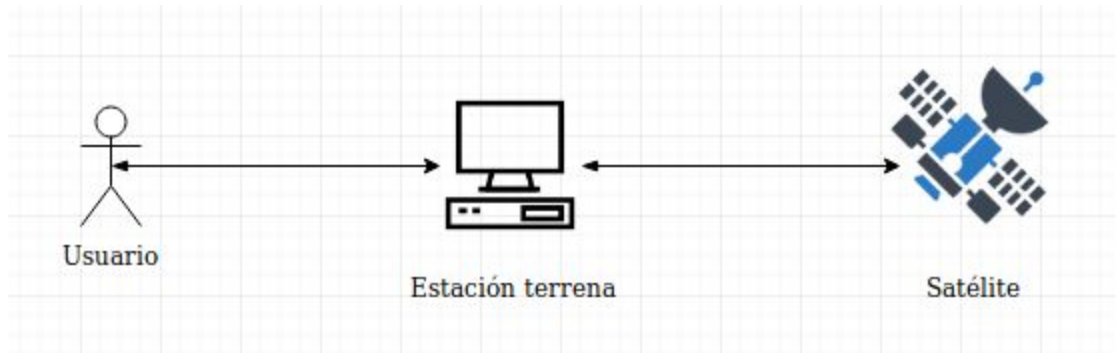


Índice

Introducción	3
Propósito	3
Ámbito del Sistema	3
Definiciones, Acrónimos y Abreviaturas	3
Referencias	3
Descripción General del Documento	3
Descripción General	4
Perspectiva del Producto	4
Funciones del Producto	4
Características de los Usuarios	4
Restricciones	4
Suposiciones y Dependencias	5
Requisitos Específicos	5
Interfaces Externas	5
Funciones	5
Requisitos de Rendimiento	5
Restricciones de Diseño	6
Atributos del Sistema	6
Diseño de solución	7
Implementación y Resultados	8
Conclusiones	13

Introducción

La estación terrena Córdoba efectúa la recepción para regular los datos satelitales del territorio nacional, la plataforma continental y los países limítrofes. Desarrolla las actividades de recepción, procesamiento, publicación y almacenamiento de la información satelital.



Propósito

El objetivo de este documento es describir el diseño y codificación de una comunicación de procesos con sockets (satélite-estación terrena). Esta comunicación ofrece una terminal con funciones implementadas en el servidor que permiten obtener información del satélite como también la transferencia de archivos.

Ámbito del Sistema

El sistema correrá en un entorno de Linux. Se busca que el servidor sea ejecutado en cualquier computadora, mientras que el cliente en una placa de desarrollo Raspberry Pi.

Definiciones, Acrónimos y Abreviaturas

No presenta ninguna acronimo y/o abreviaturas.

Referencias

[1] <http://c.conclase.net/librerias/?ansitip=™>

[2] <http://manpages.ubuntu.com/manpages/bionic/es/man1/ps.1.html>



Descripción General

Perspectiva del Producto

En este trabajo práctico para desarrollar la comunicación entre la estación terrena y un satélite, es decir la comunicación entre dos procesos, se implementará por medio de sockets.

La conexión por medio de sockets acepta diversas variantes, en este trabajo se realizó una primera versión mediante el socket UNIX y una posterior versión mediante socket INET. En ambos casos se implementaron protocolos UDP (conexión no segura) y TCP (conexión segura).

Funciones del Producto

La función principal de este producto que permita realizar la conexión, transferencia de datos satelitales y el control de los mismos entre la estación terrena y el satélite. La conexión se logra a partir de sockets TCP. Además se implementan dos funciones con este tipo de socket:

1. Update: permite una actualización del firmware del satélite.
2. Start Scanning: el satélite realizara un escaneo de toda la cara de la Tierra y posterior a esto enviará la fotografía de la misma.

Con sockets UDP se lleva a cabo la transferencia de información propia del satélite. El satélite le envía la siguiente información a la estación terrena:

- Id del satélite
- Uptime del satélite
- Versión del satélite
- Consumo de memoria
- Consumo de CPU

Características de los Usuarios

El usuario que ejecutará el servidor debe estar registrado en la base de datos y debe tener conocimiento mínimo sobre que hace cada función implementada.

Restricciones

No se garantiza seguridad ni encriptación de los datos.

Suposiciones y Dependencias

Se supone que la PC en la que correrá la estación terrena y la placa de desarrollo en la que correrá el satélite, ambas tienen instalada una distribución de Linux.

Requisitos Específicos

Interfaces Externas

Para ejecutar el satélite en la placa Raspberry Pi, es necesario una forma de comunicarse con la misma, ya sea conectando un teclado y un monitor. Otra forma de acceder es mediante ssh desde la computadora (ambas deben estar en la misma red).

Funciones

Estación terrena:

- Ofrece conexión para un único satélite, en el caso de la conexión INET a través de un puerto especificado (8182) e IP específica.
- Ofrece una autenticación de carácter local de usuario y contraseña, en caso de repetirse tres intentos sin éxito se debe cerrar el programa.
- Update: envía un archivo binario al satélite con una actualización del software, posterior a esto se debe reiniciar el satélite.

Satélite:

- Scanning: ofrece el envío de la imagen escaneada de la cara de la Tierra.
- Telemetría: Ofrece el envío de su información.

Requisitos de Rendimiento

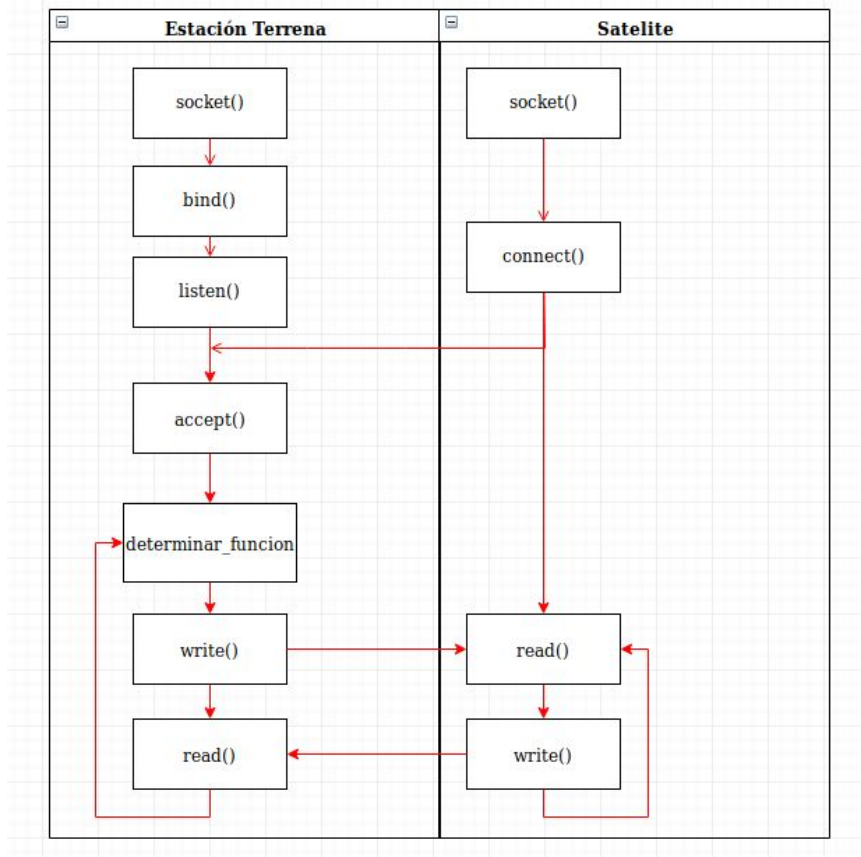
No especificado.

Restricciones de Diseño

- Utilizar la API de sockets.
- Ambos programas deben ser desarrollado en lenguaje C.
- Se realiza una primera implementación en UNIX y una posterior en INET.
- Se deben utilizar sockets orientado a conexión y sockets no orientado a la conexión.
- En la implementación INET el servidor tiene una IP y puerto fijo. El cliente debe tomar un número de puerto libre de su sistema operativo.
- Implementar un sistema de autenticación (usuario y password).
- La autenticación es de carácter local, al tercer intento debe cerrarse el programa.
- Debe incluirse un mecanismo de control y manejo de errores en todo el sistema.
- Todos los procesos deben ser mono-thread.
- Documentación del código.
- Utilizar Cppcheck y compilar con el uso de las flags de -Werror, -Wall y -pedantic.
- La transferencia de la información del satélite debe realizarse utilizando conexión no segura.
- La implementación del cliente debe ser en una placa de desarrollo.

Diseño de solución

Se inició el diseño de la solución implementando todo el sistema con sockets UNIX. Ambos programas fueron ejecutados en la misma PC en dos terminales distintas para simplicidad. Se inicia la conexión mediante un socket TCP, una vez establecida la conexión se habilita la autenticación de manera local y se imprimen las posibles funciones que el usuario puede ejecutar.



El proceso de autenticación se realiza de manera local en el servidor, el usuario ingresa su usuario y contraseña. El servidor revisa en su estructura de usuarios válidos, si los datos son correctos, se muestra un nuevo prompt con las funciones que se pueden solicitar, de lo contrario se solicita el nuevo ingreso de usuario y contraseña, si supera los 3 intentos se desconecta.

Una vez validado el usuario puede ejecutar 3 funciones ya nombradas anteriormente.

Implementación y Resultados

Se muestran los resultados de la implementación de las distintas funcionalidades del sistema tanto en UNIX como en INET.

UNIX

Autenticación: ingreso correcto del usuario y contraseña.

Se lleva a cabo la autenticación con la estructura local que se encuentra en el servidor.

```
andres@andres-X556UQK:~/Facultad/SOII/Andres/Practico/SistemasOperativos2019/S02_TP1/unix$ ./estacion ./unix
Proceso: 5261 - socket disponible: ./unix
Ingrese su usuario
nacho
Ingrese su pass
56789
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
exit en caso de salir
nacho@andres-X556UQK:~ Ingrese el mensaje a transmitir: █
```

```
strcpy(usuarios[0].user, "chiqui");
strcpy(usuarios[0].pass, "030345");
strcpy(usuarios[1].user, "andres");
strcpy(usuarios[1].pass, "01234");
strcpy(usuarios[2].user, "nacho");
strcpy(usuarios[2].pass, "56789");
strcpy(usuarios[3].user, "mary");
strcpy(usuarios[3].pass, "02468");
strcpy(usuarios[4].user, "floki");
strcpy(usuarios[4].pass, "13579");
```


Autenticación: autenticación invalida, cierre del programa.

```
andres@andres-X556UQK:~/Facultad/S0II/Andres/Practico/SistemasOperativos2019/S02_TP1/unix$ ./estacion ./unix
Proceso: 5310 - socket disponible: ./unix
Ingrese su usuario
prueba
Ingrese su pass
1
Ingrese su usuario
prueba
Ingrese su pass
2
Ingrese su usuario
prueba
Ingrese su pass
3
Se excedio de intentos, intentelo mas tarde
Fin de la conexion
```

Obtener telemetría: como se puede observar al ejecutarse esta función, el programa ya no utiliza los socket TCP si no que crea un socket UDP y por este último se lleva a cabo la transferencia de la información.

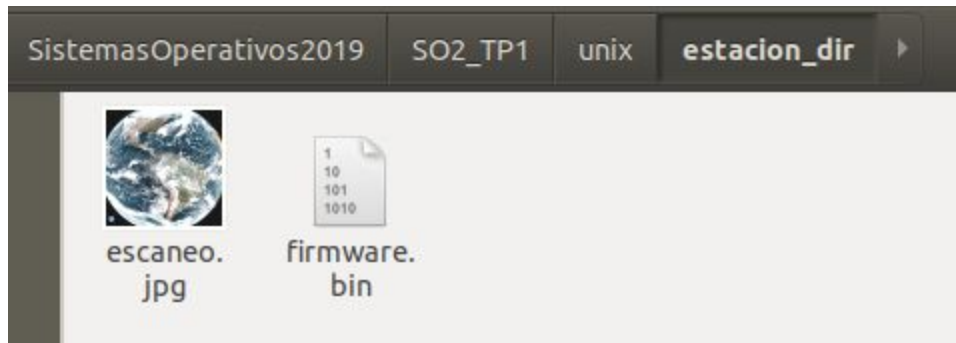
```
exit en caso de salir
floki@andres-X556UQK:~ Ingrese el mensaje a transmitir: telemetria
Socket UDP disponible: ./teludp
Id_Satelite 65096A
Uptime Satellite 0:5
Version Satellite Version0.0
Consumo CPU Satellite 0.0
Consumo memoria Satellite 4516
Socket UDP close
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
exit en caso de salir
floki@andres-X556UQK:~ Ingrese el mensaje a transmitir: █
```

Update: Como se puede observar se ejecuta primero el comando obtener telemetría y el satélite presenta una versión 0.0 y el tiempo que lleva ejecutando es de 0 minutos y 9 segundos. Al ejecutar update se produce el envío del firmware.bin al satélite el cual se reinicia. Posterior a esto se ejecuta nuevamente obtener telemetría para demostrar la actualización de la versión y el tiempo que lleva ejecutándose cambio.

```
funciones disponibles: update - scanning - telemetria
flok@andres-X556UQK:~ Ingrese el mensaje a transmitir: telemetria
Socket UDP disponible: ./teludp
Id Satellite 65096A
Uptime Satellite 0:9
Version Satellite Version0.0
Consumo CPU Satellite 0.0
Consumo memoria Satellite 4516
Socket UDP close
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
flok@andres-X556UQK:~ Ingrese el mensaje a transmitir: update
SERVIDOR: Nuevo cliente, que atiende el proceso hijo: 13572
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
flok@andres-X556UQK:~ Ingrese el mensaje a transmitir: telemetria
Socket UDP disponible: ./teludp
Id Satellite 65096A
Uptime Satellite 0:5
Version Satellite Version1.0
Consumo CPU Satellite 0.0
Consumo memoria Satellite 4516
Socket UDP close
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
flok@andres-X556UQK:~ Ingrese el mensaje a transmitir: 
```

Scanning: se observa la recepción de la imagen en la estación terrena.

```
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
flok@andres-X556UQK:~ Ingrese el mensaje a transmitir: scanning
El tamaño del paquete a recibir 77959899
```



INET:

Conexión: Se observa la conexión, esperando para la autenticación del usuario.

```
root@raspi-desktop:/home/raspi/SistemasOperativos2019/SO2_TP1/inet# ./satellite 10.0.0.3 8182
+ (raspi) 10.0.0.11 inet: estacion unix: sudo
inet: estacion - Drop-Down Terminal
```

```
andres@andres-X556UQK:~/Facultad/SOII/Andres/Practico/SistemasOperativos2019/SO2_TP1/inet$ ./estacion
Proceso: 19847 - socket disponible: 8182
Ingrese su usuario
+ (raspi) 10.0.0.11 inet: estacion unix: sudo
inet: estacion - Drop-Down Terminal
```

Telemetría: se puede observar al igual que con Unix la recepción de la información del satélite. Se llevó a cabo una captura con Wireshark para demostrar que el paquete es del tipo UDP.

```
exit en caso de salir
andres@andres-X556UQK:~ Ingrese el mensaje a transmitir: telemetria
Socket disponible: 8183
Id Satellite 65096A
Uptime Satellite 0:52
Version Satellite Version0.0
Consumo CPU Satellite 0.0
Consumo memoria Satellite 1400
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
exit en caso de salir
andres@andres-X556UQK:~ Ingrese el mensaje a transmitir:
+ (raspi) 10.0.0.11 inet: estacion unix: sudo
inet: estacion - Drop-Down Terminal
```

29	15.252299...	10.0.0.11	10.0.0.3	UDP	1066 41533 → 8183 Len=1024
----	--------------	-----------	----------	-----	----------------------------

<ul style="list-style-type: none"> Frame 29: 1066 bytes on wire (8528 bits), 1066 bytes captured (8528 bits) on interface 0 Ethernet II, Src: Raspberr_78:a5:ca (b8:27:eb:78:a5:ca), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70) Internet Protocol Version 4, Src: 10.0.0.11, Dst: 10.0.0.3 User Datagram Protocol, Src Port: 41533, Dst Port: 8183 Data (1024 bytes) 	Data: 49645f536174656c697465203635303936410a557074696d... [Length: 1024]
---	---

0020	00 03 a2 3d 1f f7 04 08 bb 42 49 64 5f 53 61 74	... Elite_Sat
0030	65 6c 69 74 65 20 36 35 30 39 36 41 0a 55 70 74	elite 65 096A-Upt
0040	69 6d 65 20 53 61 74 65 6c 69 74 65 20 30 3a 35	ime Sate lite 0:5
0050	36 0a 56 65 72 73 69 6f 6e 20 53 61 74 65 6c 69	6-Versio n Sateli
0060	74 65 20 56 65 72 73 69 6f 6e 31 2e 30 0a 43 6f	te Versi on1.0-Co
0070	6e 73 75 6d 6f 20 43 50 55 20 53 61 74 65 6c 69	nsuno CP U Sateli
0080	74 65 20 30 2e 30 0a 43 6f 6e 73 75 6d 6f 20 6d	te 0.0-C onsumo m
0090	65 6d 6f 72 69 61 20 53 61 74 65 6c 69 74 65 20	emoria S atelite
00a0	31 34 30 30 00 00 00 00 00 00 00 00 00 00 00	1400....

Update: se lleva a cabo la actualización del firmware del satélite. Se observa la actualización en la versión del mismo.

```

exit en caso de salir
andres@andres-X556UQK:~ Ingrese el mensaje a transmitir: update
FE0F 0
Read size 11
n 11
Numero de paquete: 1
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
exit en caso de salir
andres@andres-X556UQK:~ Ingrese el mensaje a transmitir: telemetria
Socket disponible: 8183
Id Satellite 65096A
Uptime Satellite 0:2
Version Satellite Version1.0
Consumo CPU Satellite 0.0
Consumo memoria Satellite 1400
Ingrese la funcion que desea ejecutar
Funciones disponibles: update - scanning - telemetria
exit en caso de salir
andres@andres-X556UQK:~ Ingrese el mensaje a transmitir:
+ (raspi) 10.0.0.11 inet:estacion unix:sudo

```

Scanning: se realizó la captura en wireshark para demostrar que el envío del mismo se realiza por TCP.

707	79.654400...	10.0.0.11	10.0.0.3	TCP	1514 57046 → 8182 [PSH, ACK] Seq=141912 Ack=6642336 Win=0 Len=1448
708	79.657358...	10.0.0.11	10.0.0.3	SSH	150 Server: Encrypted packet (len=84)
709	79.657529...	10.0.0.3	10.0.0.11	TCP	66 42336 → 22 [ACK] Seq=1 Ack=12401 Win=0 Len=0

<ul style="list-style-type: none"> Frame 707: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0 Ethernet II, Src: Raspberr_78:a5:ca (b8:27:eb:78:a5:ca), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70) Internet Protocol Version 4, Src: 10.0.0.11, Dst: 10.0.0.3 Transmission Control Protocol, Src Port: 57046, Dst Port: 8182, Seq: 141912, Ack: 2055, Len: 1448 Data (1448 bytes) 	Data: c7ea2029de37ded231f1d3ea7af18fd2d7d47f99bfd3eef1... [Length: 1448]
--	---

0000	3c a0 67 3f 0e 70 b8 27 eb 78 a5 ca 08 00 45 00	<.g?.p.'-X...E.
0010	05 dc 0f 75 40 00 40 06 11 9a 0a 00 00 0b 0a 00	...u@.@:.....
0020	00 03 de d6 1f f6 d5 f5 b5 71 d6 3c 4b d3 80 18-q-<K...

Conclusiones

Al finalizar este trabajo práctico se pudo consolidar lo aprendido en las clases teóricas sobre los conceptos de sockets y como estos son aplicados en las comunicaciones entre procesos. Se pudieron llevar a la práctica estos conceptos logrando cumplir los objetivos planteados en la consigna.

Se obtuvieron conocimiento de como operan llamadas al sistema del kernel de Linux. Se presentaron algunas dificultades en el momento del envio y recepcion de archivos, como también al momento de obtener información sobre el sistema, pero se pudieron superar estos obstáculos y lograr el objetivo.

Por último y no menor, obtuve conocimiento sobre el manejo de un nuevo embebido al cual no había manipulado nunca.