

Alexander Duran Martínez

Andrew Araya Vega

Benjamín Calvo De León

TECNOLÓGICO DE COSTA RICA
ESCUELA DE COMPUTACION
ADMINISTRACION DE TECNOLOGÍAS
DE INFORMACIÓN
LENGUAJES DE PROGRAMACION
PROFESOR: ANDREI FUENTES LEIVA
TAREA PROGRAMADA #1 LENGUAJE C

Tabla de Contenido

Descripción del problema	2
Cual es la situacion que se presenta:	2
Solucion al problema:	2
Diseño del programa	2
Decisiones de diseño:	2
Diagrama lógico.	3
Análisis de Resultados.	4
Objetivos alcanzados:	4
Objetivos no alcanzados:	5
Librerías usadas	5
Manual de usuario	5
Servidor:	6
Cliente:	6
Conclusión personal.	7

DESCRIPCIÓN DEL PROBLEMA

Cual es la situacion que se presenta:

Se presenta una tarea programada en lenguaje C en el sistema operativo Linux, se pide hacer un programa que sea capaz de enviar y recibir archivos de cualquier tipo en al menos, una red local, o enviar y recibir archivos al mismo tiempo, de un tamaño máximo predefinido por el programador, esto implica una serie de pasos y eventos que llevarán a un resultado exitoso.

Solucion al problema:

- Primero se debe plantear un proceso que deje ver cuáles son los verdaderos requisitos del programa y sus problemas, en el primer paso se notamos que al ser un envío y recibo de archivos se deben usar sockets. Esto procede a una investigación de cómo usarlos, cuáles requisitos necesitarán y cuáles librerías deberán ser incluidas haciendo inclusión de código.
- A partir de los sockets se debe crear un programa que pueda enviar los archivos y que pueda recibirlos, además de que pueda realizar las dos funciones al mismo tiempo, para esto será necesario usar la función fork (). La función fork (), permite dividir un proceso en dos, para presentar un ejemplo sencillo se puede ver como un tenedor, que viene desde una sola parte y que se divide en varios objetos.

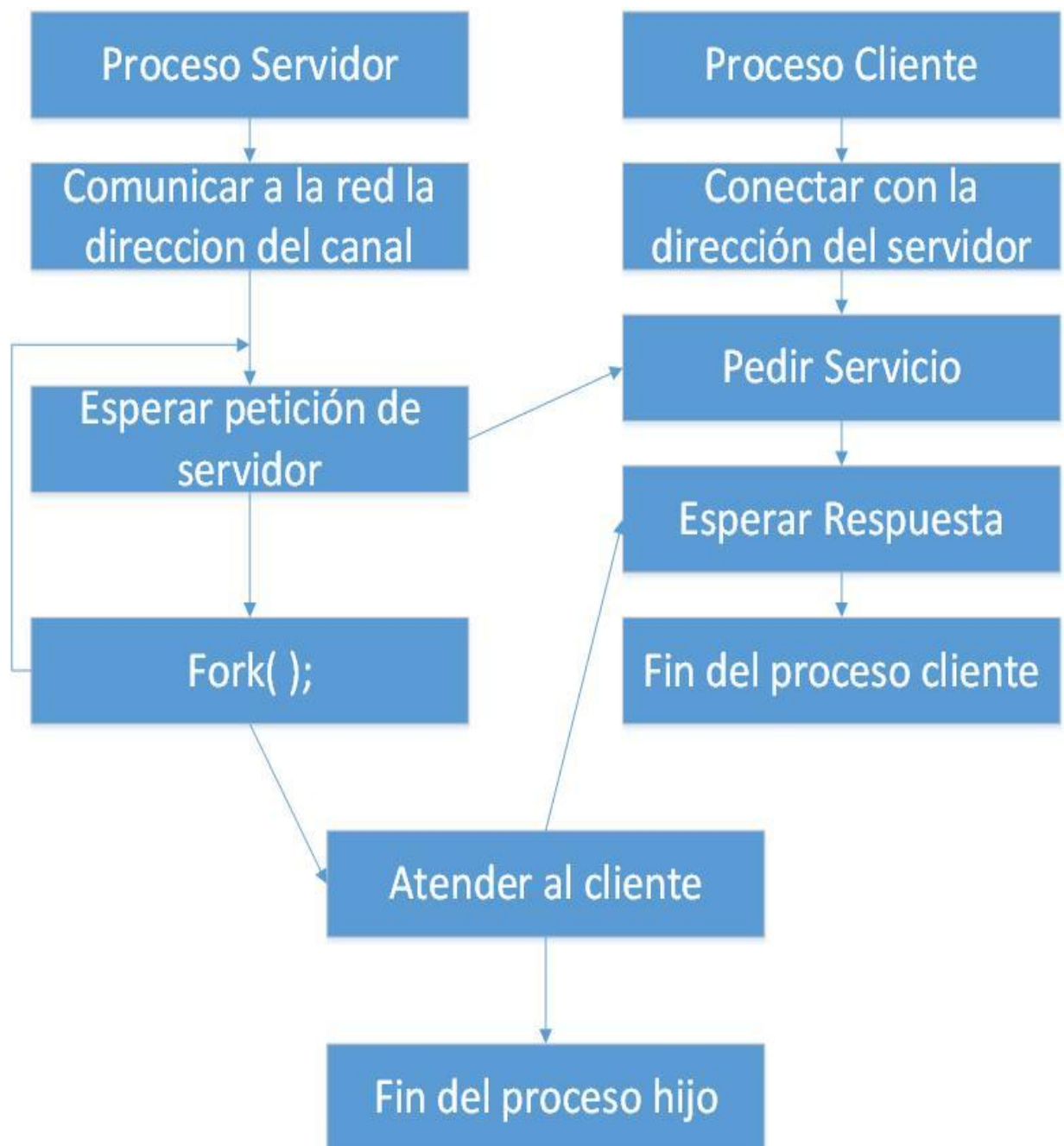
DISEÑO DEL PROGRAMA

Decisiones de diseño:

- Luego de haber pensado en los elementos necesarios para realizar dicho programa, se decide usar "Geany", un software que soporta el desarrollo de varios de lenguajes en Linux tales como Python, java, C++ y otro.
- Para diseñar el programa se reparten las funciones entre los tres miembros, uno investiga sobre la funcion fork(), otro sobre el envío de archivos y otro sobre la recepción de archivos, sin embargo, a la hora de terminar la investigación se inicializa el proceso de programación, para esto utilizamos un cable de red para poder realizar la red local.

DIAGRAMA LÓGICO.

- De acuerdo a este diagrama, logramos proponer cuales eran los diferentes aspectos que debíamos cumplir del programa y además poder hacer una estructura o guía de lo que debíamos realizar paso a paso.



ANÁLISIS DE RESULTADOS.

Objetivos alcanzados:

- Dentro de los objetivos se logró realizar el envío de archivos alambrado, es decir en la misma maquina los dos programas, además, el envío y recibo de archivos en dos computadoras conectadas mediante una red local y de forma inalámbrica.

Objetivos no alcanzados:

- No se logra hacer el `fork()`, es decir, la bifurcación de procesos, ya que no lo pudimos completar, fue en realidad un proceso difícil y no logramos tener éxito, esto debido a que nos fue imposible insertar la función por falta de conocimiento y desconocimiento del error dentro del código.

LIBRERÍAS USADAS

- `#include <netinet/in.h>`
- `#include <stdio.h>`
- `#include <stdlib.h>`
- `#include <unistd.h>`
- `#include <string.h>`
- `#include <sys/types.h>`
- `#include <netinet/in.h>`
- `#include <string.h>`
- `#include <arpa/inet.h>`

MANUAL DE USUARIO

- Iniciar el Sistema operativo Linux.
- Abrir la terminal, para realizar se puede utilizar el comando (`ctrl+alt+t`):

A screenshot of a terminal window with a dark purple background. The prompt 'araya@ubuntu:~\$' is visible at the top left, followed by a white cursor. The rest of the terminal area is empty.

- Se busca la dirección donde se encuentra el programa.

- Se compila el archivo, basta utilizar el comando “gcc” y el nombre del archivo junto con la extensión “.c”, de la siguiente forma:

```
araya@ubuntu: ~  
araya@ubuntu:~$ ls  
Descargas  eclipse          Imágenes          Plantillas  Ubuntu One  
Documentos Escritorio      lenguajes-de-programacion Público      Videos  
Dropbox    examples.desktop Música            Servidor.c  workspace  
araya@ubuntu:~$ gcc Servidor.c  
araya@ubuntu:~$ ls  
a.out      eclipse          lenguajes-de-programacion Servidor.c  
Descargas  Escritorio      Música            Ubuntu One  
Documentos examples.desktop Plantillas        Videos  
Dropbox    Imágenes        Público           workspace  
araya@ubuntu:~$
```

Se inicializa el archivo “a.out”, para esto, es necesario ejecutarlo, y darle los parámetros de IP, de servidor que envía/recibe y el que queda en espera, de la siguiente forma:

Servidor:

```
araya@ubuntu: ~  
araya@ubuntu:~$ ls  
Descargas  eclipse          Imágenes          Plantillas  Ubuntu One  
Documentos Escritorio      lenguajes-de-programacion Público      Videos  
Dropbox    examples.desktop Música            Servidor.c  workspace  
araya@ubuntu:~$ gcc Servidor.c  
araya@ubuntu:~$ ls  
a.out      eclipse          lenguajes-de-programacion Servidor.c  
Descargas  Escritorio      Música            Ubuntu One  
Documentos examples.desktop Plantillas        Videos  
Dropbox    Imágenes        Público           workspace  
araya@ubuntu:~$ ./a.out 55000
```

Después escribiremos el nombre del archivo que queremos enviar.

Cliente:

```
Uso:  
1) Digite la dirección IP del cliente, 2) El número de puerto del cliente y 3) El  
número de puerto de escucha de éste equipo  
araya@ubuntu:~$ ./a.out 192.168.1.1 55000 56000  
  
Se ha creado el socket. Puerto #55510  
Se ha enlazado el socket  
Esperando una conexión...  
█
```

CONCLUSIÓN PERSONAL.

A nuestro parecer nos resulta bastante buena la tarea programada, puso a prueba las habilidades de investigación, de programación y sobre todo de unión de equipo, para poder cumplir los diferentes aspectos del programa. Además, en los comentarios que hacíamos, es de los pocos programas en los que podemos ver una verdadera función aplicada a la vida real, es decir, enviar y recibir archivos es prácticamente un correo, algo que es bastante útil en la actualidad, aunque tuvimos ciertos problemas respecto al github y al uso de la función fork, logramos cumplir las expectativas y realizar el programa como lo deseábamos. Tuvimos un pequeño problema respecto a la semana santa ya que no pudimos reunirnos y el adelanto que llevábamos no nos sirvió y más bien tuvimos cierto atraso que nos afectó y nos hizo tomar más tiempo del que se había planeado.