Proyecto: Monitoreo de Sensores
Sistemas Operativos
2024

Juan David Rincón Poveda
Ingeniería de Sistemas

Pontificia Universidad Javeriana

Bogotá

Juan Felipe Morales Espitia
Ingeniería de Sistemas

Pontificia Universidad Javeriana

Bogotá

***Resumen*: El proyecto "Monitoreo de Sensores" tiene como objetivo desarrollar un sistema de monitoreo de la calidad del agua mediante la simulación de sensores de temperatura y pH. Este sistema utiliza herramientas de comunicación y sincronización de procesos e hilos, como pipes nominales y semáforos, para coordinar la transferencia de datos entre los diferentes componentes del sistema. El monitoreo de la calidad del agua es crucial para detectar cambios y tomar acciones correctivas, y este proyecto proporciona una solución eficiente y escalable para este propósito.**

Palabras clave: temperatura, sensores, búffer, hilos, mecanismos de comunicación.

I. Introducción

El agua es un recurso vital en nuestro planeta, y su calidad es fundamental para diversos usos, como la agricultura y la generación de energía. En este contexto, es importante monitorear constantemente la calidad del agua para detectar cambios y tomar medidas apropiadas. El proyecto "Monitoreo de Sensores" aborda esta necesidad mediante la simulación de sensores de temperatura y pH, que proporcionan mediciones precisas de estos parámetros. El sistema desarrollado utiliza pipes nominales y semáforos para facilitar la comunicación y sincronización entre los diferentes componentes del sistema, garantizando un monitoreo efectivo y en tiempo real de la calidad del agua.

En este proyecto realizaremos un sistema sencillo donde se simula la medición de dos parámetros de una reserva de agua a través de sensores: PH y temperatura. Estos parámetros se enviarán a un proceso monitor, que los almacenará adecuadamente y avisará al usuario si se genera alguna alerta con los indicadores medidos.

II. Diseño

El sistema consta de dos componentes principales: los sensores y el monitor. Los sensores simulan la medición de la temperatura y el pH del agua, mientras que el monitor recibe estas mediciones, las almacena y genera alertas en caso de detectar valores fuera de los rangos aceptables. La comunicación entre los sensores y el monitor se realiza a través de pipes nominales, mientras que la sincronización de los procesos.

A continuación, se explica cada uno de los componentes de la arquitectura de software

(figura 1):

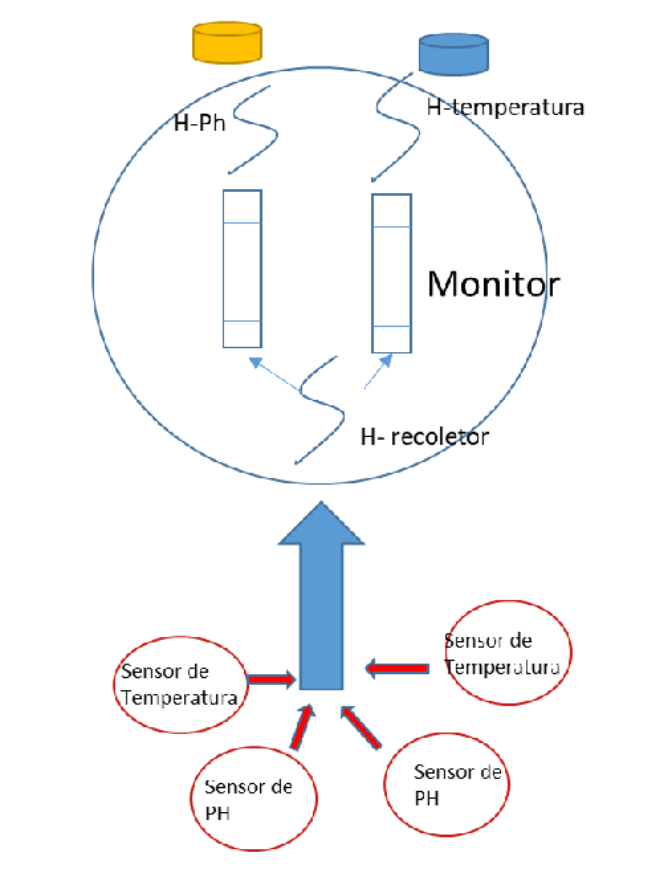

Fig. 1. Procesos/hilos y mecanismos de comunicación

- **Sensores:** Los sensores que miden la temperatura y el PH serán simulados por procesos. Los procesos sensores se invocarán desde el shell de la siguiente forma (MakeFile):

$ ./sensor –s tipo-sensor –t tiempo –f archivo –p pipe nominal

Donde:

- **tipo_sensor:** Dado que cada proceso sensor solo reportará los valores de una variable, se utilizará está parámetro para especificar si el proceso reportará PH o temperatura. Se utilizará el número 1 para temperatura y el número 2 para el PH.
- **tiempo:** este parámetro indica cada cuanto tiempo (t) se va a enviar la medición del proceso sensor al proceso monitor. Los valores de las variables se van leyendo desde un archivo. Cada proceso sensor, lee una medida del archivo, espera t segundos y luego envía la siguiente medición al monitor.

- **archivo:** es el nombre del archivo con medidas de temperatura o de PH, según sea el caso. El archivo tendrá tres tipos de valores. a) Valores correctos y dentro de los rangos establecidos en la tabla 1. b) Valores positivos, pero por fuera de los rangos establecidos en la tabla (estos valores generarán alertas) c) Valores negativos: estos valores se consideran erróneos, por tanto, serán descartados al ser recibidos por el monitor.

  - **Datos.txt:** Acá se encuentran los datos que tomara cada sensor tanto el de temperatura como el de PH :

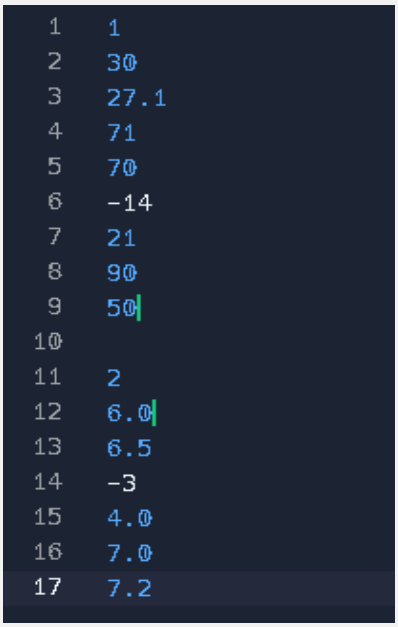

Fig. 2. Representación datos.txt

  - **Pipe-nominal:** Es el nombre del pipe nominal que permite la comunicación entre los sensores y el monitor.

    Además, se implementaron bandera **-s** y **-t** como parámetros de nuestro MakeFile a la hora de compilar nuestro código.

    $ ./sensor –s 1 –t 3 –f datos –p pipe1

    $ ./sensor –t 10 –p mypipe –s 2 –f file-input.txt

Parámetro	Valor mínimo	Valor máximo
Temperatura	**20°C**	**31.6°C**
PH	**6.0**	**8.0**

Fig. 3. Rangos de los parámetros de calidad.

  - **Monitor:** Es el proceso que recibe las medidas de los sensores. El monitor estará internamente conformado por tres hilos. La función de cada uno de los hilos se describe a continuación:

    - **H-recolector:** Recibe los dos tipos de mediciones del pipe nominal y las coloca en uno de los dos búferes dependiendo del tipo de medida (PH o Cuando el recolector detecte que no hay ningún sensor conectado, esperará 10 segundos antes de enviar temperatura). Si le medida recibida por el pipe es errónea (un valor negativo), se descarta y no se coloca en ninguno de los dos búferes; en este caso el recolector debe imprimir un mensaje por la consola. un mensaje al resto de los hilos indicando que ya terminó el envío de indicadores. Una vez colocado este mensaje en cada buffer, el hilo recolector elimina el pipe nominal, imprime un mensaje de finalización del procesamiento de medidas y termina.

- **H-ph:** Este hilo recoge las medidas que son colocadas en su buffer (medidas de PH). Todas las medidas recibidas se escribirán en el archivo file-ph, cuyo nombre recibe el monitor como argumento de entrada. El hilo H-ph, antes de escribir una determinada medida en el archivo, le anexará la hora actual. Si la media está fuera de los rangos, el hilo imprime un mensaje de alerta por la cónsola. Cuando este hilo recibe un mensaje de finalización por parte del hilo recolector, cierra el archivo y termina.
- **H-temperatura:** Este hilo recoge las medidas que son colocadas en su buffer (medidas de temperatura). Todas las medidas recibidas se escriben en el archivo file-temp, cuyo nombre recibe el monitor como argumento de entrada. A cada medida se le anexa la hora actual antes de escribirla. Si la media está fuera de los rangos, el hilo imprime un mensaje de alerta por la consola. Cuando este hilo recibe un mensaje de finalización por parte del hilo recolector, cierra el archivo y termina.

Finalmente, el proceso monitor se invocará desde el shell con los siguientes parámetros:

./monitor –b tam_buffer –t file-temp –h file-ph -p pipe-nominal

Donde:

tam-buffer: es el tamaño de los búferes donde el H-recolector colocará las medidas.

file_temp: es el nombre del archivo de texto donde el hilo H-temperatura colocará las mediciones de temperatura recibidas.

file_ph: es el nombre del archivo de texto donde el hilo H-Ph colocará las mediciones de PH recibidas.

pipe-nominal: es el nombre del pipe nominal que permite la comunicación entre los sensores y el monitor.

III. Resultados Obtenidos

Durante el desarrollo del proyecto, se implementó con éxito un sistema funcional de monitoreo de sensores. Se realizaron pruebas exhaustivas para verificar la precisión y confiabilidad de las mediciones, y se demostró que el sistema es capaz de detectar y manejar valores fuera de los rangos aceptables de temperatura y pH. Además, se logró una comunicación eficiente entre los diferentes componentes del sistema, garantizando un monitoreo continuo y en tiempo real de la calidad del agua.

SENSOR 1:

Datos.txt

```
1
30
27.1
71
70
-14
21
90
50

2
6.0
6.5
-3
4.0
7.0
7.2
```

MakeFile:

SO_proyecto > Makefile

```
gcc = gcc
CXXFLAGS = -Wall -Wextra -Iinclude -lpthread

all: sensor monitor

sensor: sensor.c
  $(gcc) $(CXXFLAGS) $^ -o $@

monitor: monitor.c buffer.c
  $(gcc) $(CXXFLAGS) $^ -o $@

run: run_sensor run_monitor

run_sensor: sensor
  @echo "Ejecutando sensor en segundo plano..."
  @./sensor -t 3 -s 1 -f datos.txt -p pipe1 &

run_monitor: monitor
  @echo "Ejecutando monitor en segundo plano..."
  @./monitor -b 10 -t file-temp.txt -h file-ph.txt -p pipe1 &
```

Se modifica la bandera -s 1 (Temp) para indicar que se va a usar el sensor 1 lo que resultaría la compilación en:

```
make run
Ejecutando sensor en segundo plano...
Failed to open pipe: pipe1, retrying...
Ejecutando monitor en segundo plano...
Buffer size: 10
Pipe name: pipe1
File temp: file-temp.txt
File ph: file-ph.txt
Pipe created successfully: pipe1
~/Proyecto-Operativos/SO_proyecto$ Pipe opened successfully: pipe1
Iniciando sección de datos de temperatura...
Enviando datos al pipe: 30
Enviando datos al pipe: 27
Enviando datos al pipe: 71
Enviando datos al pipe: 70
Enviando datos al pipe: -14
Enviando datos al pipe: 21
Enviando datos al pipe: 90
Enviando datos al pipe: 50
Fin de la sección de datos.
Pipe opened successfully: pipe1
```

Por lo que finalmente escribiría en el archivo file-temp.txt lo siguiente:

SO_proyecto > file-temp.txt

```
1 20:05:18
30 20:05:21
27 20:05:25
71 20:05:30
70 20:05:33
21 20:05:38
90 20:05:44
50 20:05:50
```

SENSOR 2:

Datos.txt

```
1
30
27.1
71
70
-14
21
90
50

2
6.0
6.5
-3
4.0
7.0
7.2
```

MakeFile:

SO_proyecto > Makefile

```
gcc = gcc
CXXFLAGS = -Wall -Wextra -Iinclude -lpthread

all: sensor monitor

sensor: sensor.c
  $(gcc) $(CXXFLAGS) $^ -o $@

monitor: monitor.c buffer.c
  $(gcc) $(CXXFLAGS) $^ -o $@

run: run_sensor run_monitor

run_sensor: sensor
  @echo "Ejecutando sensor en segundo plano..."
  @./sensor -t 3 -s 2 -f datos.txt -p pipe1 &

run_monitor: monitor
  @echo "Ejecutando monitor en segundo plano..."
  @./monitor -b 10 -t file-temp.txt -h file-ph.txt -p pipe1 &
```

Se modifica la bandera -s 2 (Ph) para indicar que se va a usar el sensor 2 lo que resultaría la compilación en:

```
make run
Ejecutando sensor en segundo plano...
Failed to open pipe: pipe1, retrying...
Ejecutando monitor en segundo plano...
Buffer size: 10
Pipe name: pipe1
File temp: file-temp.txt
File ph: file-ph.txt
Pipe created successfully: pipe1
~/Proyecto-Operativos/SO_proyecto$ Pipe opened successfully: pipe1
Pipe opened successfully: pipe1
Iniciando sección de datos de pH...
Enviando datos al pipe: 6.0
Enviando datos al pipe: 6.5
Enviando datos al pipe: -3
Enviando datos al pipe: 4.0
Enviando datos al pipe: 7.0
Enviando datos al pipe: 7.2Failed to open pipe: pipe1, retrying...
```

Por lo que finalmente escribiría en el archivo file-ph.txt lo siguiente:

SO_proyecto > file-ph.txt

```
1  6 20:01:09
2  6.5 20:01:12
3  4 20:01:18
4  7 20:01:21
5  7.2 20:01:24
6
```

IV. Conclusiones

El proyecto de monitoreo de sensores ha sido un paso importante hacia la implementación de sistemas eficientes para la vigilancia y control de la calidad del agua. A través de la simulación de sensores de temperatura y pH, hemos desarrollado un sistema que puede proporcionar mediciones precisas y en tiempo real de estos parámetros críticos.

Durante el proceso de desarrollo, enfrentamos varios desafíos que nos permitieron aprender y aplicar una amplia gama de conceptos y técnicas relacionadas con la programación de sistemas operativos. Desde la implementación de pipes nominales y

semáforos hasta la coordinación de procesos e hilos, cada aspecto del proyecto nos proporcionó una valiosa experiencia en el diseño y la implementación de sistemas distribuidos.

Uno de los logros más significativos del proyecto fue la capacidad de detectar y manejar valores anómalos de temperatura y pH. Implementamos mecanismos de alerta que notifican al usuario cuando se detectan valores fuera de los rangos aceptables, lo que permite una acción rápida y efectiva para corregir cualquier problema potencial.

Además, la implementación de búferes acotados y el patrón productor/consumidor nos permitieron garantizar una comunicación fluida entre los diferentes componentes del sistema. Esto aseguró que las mediciones se transmitieran de manera eficiente y sin pérdida de datos, incluso en entornos de alta carga.

En resumen, el proyecto de monitoreo de sensores nos ha brindado una experiencia invaluable en el desarrollo de sistemas operativos complejos. Hemos adquirido habilidades prácticas en el diseño, implementación y prueba de sistemas distribuidos, así como un profundo entendimiento de los principios fundamentales que subyacen a su funcionamiento. Estamos seguros de que esta experiencia nos servirá como base sólida para abordar desafíos aún más grandes en el futuro.