

Trabajo Práctico 1

Sistemas Operativos (72.11)



Juan Godfrid - 56609
Pablo Radnic - 57013
Joaquín Ormachea -
Francisco Delgado -

27 de marzo de 2018

Índice general

1. Diseño	2
1.1. Diseño del Sistema	2
1.2. IPCs	2
1.2.1. Message Queue	2
1.2.2. Señales	2
2. Utilización	3
2.1. Compilación y Ejecución	3
2.2. Proceso Vista	3
3. Problemas y Resoluciones	4
3.1. Proceso Maestro	4
3.2. Proceso Esclavo	4
3.3. Proceso Vista	4
4. Limitaciones	5
4.1. Limitaciones del Sistema	5
4.2. Limitaciones de la Implementación	5
5. Bibliografía	6
5.1. Extractos de código	6

Capítulo 1

Diseño

1.1. Diseño del Sistema

dshfdshfdshfds

1.2. IPCs

1.2.1. Message Queue

El sistema utiliza dos *message queues* del tipo POSIX (son más modernas que sus contrapartes SYSTEM V) en la intercomunicación de los procesos maestro y esclavo. El primero se utiliza para enviar los archivos que se requieren *hashear* hacia los esclavos, mientras que el segundo se utiliza para enviar los *hashes* ya procesados devuelta al proceso maestro. La utilidad de dicho IPC es una multi-causalidad. Al ser una cola, el *message queue* asegura el comportamiento LIFO, por lo que es ideal para enviar los *hashes* de vuelta al proceso maestro, ya que asegura que el orden de llegada corresponde al orden de procesamiento. El *message queue* también resulta adecuado para enviar los nombres de los archivos a los esclavos ya que los mensajes sirven como instrucciones atómicas: Los esclavos leen las instrucciones de la cola, cuando terminan, leen otra instrucción, y cuando está vacía la cola significa que el procesamiento concluyó. Todo esto se realiza sin gastar un solo ciclo de procesador del proceso maestro.

1.2.2. Señales

Capítulo 2

Utilización

2.1. Compilación y Ejecución

El sistema posee un archivo de tipo *GNU make* que discrimina en la compilación del código fuente en base a qué proceso debería pertenecer, de este modo logra compilar separadamente los archivos binarios de manera autónoma. El programa debe recibir como argumento por línea de comandos una lista de archivos a procesar, también puede recibir una expresión *bash* para los archivos (por ejemplo: `./*.c`). Un ejemplo de uso del programa es el siguiente:

```
/SO-TP1$ make  
/SO-TP1$ ./Binaries/run main.c
```

2.2. Proceso Vista

El *make* compila también de manera separada el proceso vista, para ejecutarlo se debe enviar el siguiente comando:

Capítulo 3

Problemas y Resoluciones

3.1. Proceso Maestro

asdf

3.2. Proceso Esclavo

sdfasdf

3.3. Proceso Vista

asdfadsf Dummy text

Capítulo 4

Limitaciones

4.1. Limitaciones del Sistema

asdfasldjfasd

4.2. Limitaciones de la Implementación

asdfasdf Dummy text

Capítulo 5

Bibliografía

5.1. Extractos de código

Dummy text