

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Departamento de ciencias computacionales.

Ingeniería en computación



Seminario de Solución de Problemas de Sistemas Operativos - D01.

Violeta del Rocío Becerra Velázquez

“Khbit”

12/02/2023

Olguín Hernández Jair Benjamín

217439707

Índice

Contenido.....	3
1. ¿Qué significa JCL?:	3
2. Escriba la diferencia entre el procesamiento por lotes y el de procesamiento por lotes con multiprogramación.	3
3. Según la evolución del Sistema Operativo, liste lo que siguió al proceso por lotes con multiprogramación.	3
4. Escriba una de las utilidades de la interrupción int86 en C.	4
5. ¿Para qué sirve la función Kbhit?	4
6. Investigue el equivalente de Kbhit (utilizado en c) en otros dos lenguajes de programación y escríbalos.	4
7. ¿Qué lenguaje de programación es el que utiliza y por qué?	5
Conclusión.....	5
Bibliografía.....	5

Contenido.

1. ¿Qué significa JCL?:

JCL de las siglas Job Control Language, se trata de un lenguaje de programación utilizado en sistemas operativos IBM Mainframe para especificar y controlar tareas que deben ser ejecutadas por el sistema. Con JCL, los usuarios pueden especificar los detalles de un trabajo, como los datos de entrada, los programas que deben ser ejecutados y las operaciones de salida, como impresión o almacenamiento en disco. JCL es un lenguaje muy importante en el ámbito de la informática mainframe y sigue siendo ampliamente utilizado en muchos entornos empresariales grandes y críticos.

2. Escriba la diferencia entre el procesamiento por lotes y el de procesamiento por lotes con multiprogramación.

El procesamiento por lotes es un enfoque de procesamiento de datos en el que un gran número de tareas o trabajos se agrupan y se procesan en un orden determinado. En un sistema de procesamiento por lotes, los trabajos se envían a una cola y se ejecutan secuencialmente sin interacción con el usuario. Esto permite que el sistema utilice de manera eficiente los recursos, como la CPU, la memoria y el disco, para procesar una gran cantidad de datos de manera automatizada. El procesamiento por lotes con multiprogramación, por otro lado, es una técnica avanzada de procesamiento por lotes en la que se pueden ejecutar varios trabajos al mismo tiempo. El sistema operativo divide la CPU y otros recursos entre los trabajos y los ejecuta simultáneamente, lo que aumenta la eficiencia y reduce el tiempo de procesamiento. La multiprogramación también permite que el sistema responda a múltiples requerimientos al mismo tiempo y ofrezca una experiencia más interactiva para el usuario. El procesamiento por lotes es un enfoque básico de procesamiento de datos en el que los trabajos se ejecutan secuencialmente, mientras que el procesamiento por lotes con multiprogramación es una técnica avanzada que permite ejecutar varios trabajos al mismo tiempo y mejora la eficiencia y la interacción con el usuario.

3. Según la evolución del Sistema Operativo, liste lo que siguió al proceso por lotes con multiprogramación.

Después del procesamiento por lotes con multiprogramación, la evolución del sistema operativo incluyó los siguientes desarrollos:

- Sistemas operativos de tiempo compartido: Los sistemas operativos de tiempo compartido permiten a varios usuarios compartir recursos del sistema, como la CPU y la memoria, al mismo tiempo. Esto permitió una mayor interacción y colaboración entre los usuarios.
- Sistemas operativos distribuidos: Los sistemas operativos distribuidos permiten a los usuarios acceder y utilizar recursos de diferentes computadoras en una red. Esto permitió una mayor escalabilidad y eficiencia en la gestión de grandes cantidades de datos.

- Sistemas operativos basados en GUI: Los sistemas operativos basados en GUI introdujeron una interfaz gráfica de usuario para la interacción con el sistema operativo. Esto permitió una experiencia más intuitiva y fácil de usar para el usuario.
- Sistemas operativos basados en objetos: Los sistemas operativos basados en objetos utilizan la programación orientada a objetos para simplificar la gestión de los recursos del sistema. Esto permitió una mayor flexibilidad y escalabilidad en la gestión de grandes cantidades de datos.

4. Escribe una de las utilidades de la interrupción int86 en C.

La interrupción INT 86 en C se utiliza en programación de bajo nivel para acceder a servicios de hardware y de sistema operativo. Dentro de las utilidades de la interrupción esta:

- Acceso a dispositivos de entrada y salida: La interrupción INT 86 permite acceder a dispositivos de entrada y salida, como el teclado, el ratón, el disco y la pantalla.
- Acceso a servicios de sistema operativo: La interrupción INT 86 permite acceder a servicios de sistema operativo, como la creación y eliminación de archivos, la manipulación de directorios y la gestión de memoria.
- Acceso a la memoria de vídeo: La interrupción INT 86 permite acceder a la memoria de vídeo y controlar la forma en que se muestra la información en la pantalla.
- Acceso a servicios de reloj y tiempo: La interrupción INT 86 permite acceder a servicios de reloj y tiempo, como la gestión del tiempo y las fechas.
- Acceso a servicios de red: La interrupción INT 86 permite acceder a servicios de red, como la gestión de direcciones IP y la transmisión de datos.

5. ¿Para qué sirve la función Kbhit?

La función kbhit es una función de consola en C que se utiliza para determinar si una tecla ha sido presionada por el usuario en el teclado. Esta función se utiliza a menudo en programación de consola para permitir una entrada sin bloqueo, lo que significa que el programa puede continuar ejecutándose mientras espera la entrada del usuario. La función kbhit devuelve un valor booleano que indica si se ha presionado alguna tecla en el teclado. Si se ha presionado una tecla, la función devuelve true y si no se ha presionado una tecla, devuelve false. De esta manera, un programa puede esperar la entrada del usuario sin bloquear la ejecución del programa.

6. Investigue el equivalente de Kbhit (utilizado en c) en otros dos lenguajes de programación y escríbalos.

En C++ se utiliza la misma función de Kbhit, en Python se puede utilizar las librerías termios y fcntl para configurar el flujo de entrada de teclado (sys.stdin) como no bloqueante, y utilizar un bucle while para verificar si hay algún carácter disponible para leer. Y en java tenemos "java.io.IOException", que esta devuelve un valor booleano que indica si hay algún carácter disponible para leer en el flujo de entrada de teclado

(System.in). Y también se puede utilizar un bucle while que se ejecute mientras no haya ningún carácter disponible.

7. ¿Qué lenguaje de programación es el que utiliza y por qué?

Para distintos programas donde pueda trabajar en consola o no necesite de una interfaz dedicada especialmente a usuarios C++, debido a las facilidades que tengo para resolver problemas. Pero si necesito crear algún programa que necesite de una interfaz utilizo Python debido a que con este puedo hacer interfaces de manera mas sencillas.

Conclusión.

Esta investigación me resulto más fácil de realizar ya que con la anterior tenía conocimientos sobre el procesamiento en lotes y llegue a leer de la multiprogramación, entonces el investigar y entender sobre los JCL fue un poco más fácil. Aunque personalmente lo que me pareció mas interesante fue el buscar sobre el funcionamiento de Kbhit. Ya que nunca había llegado a utilizar algún programa que tuviera que detectar el si una tecla es pulsada. Que sabiendo que es lo que vamos a utilizar para el próximo programa me tocara buscar y conocer más de su funcionamiento y del equivalente de Python.

Bibliografía

Definición del JCL. (s. f.). © Copyright IBM Corporation 2011, 2016. © Copyright HCL Technologies Limited 2016, 2017. <https://www.ibm.com/docs/es/workload-automation/9.4.0?topic=zos-defining-jcl>

Hansen, P. B. (2001). *The Evolution of Operating Systems*. SpringerLink. https://link.springer.com/chapter/10.1007/978-1-4757-3510-9_1?error=cookies_not_supported&code=38aed46c-ca9a-4204-aeca-cef7d85eb076

Int86 function in C| System interrupt, Union REGS, Int86x, Intdos, intdosx. (s. f.). <https://www.equestionanswers.com/c/c-int86-dos-bios-system-interrupts.php>

Hansen, P. B. (2001). *The Evolution of Operating Systems*. SpringerLink. https://link.springer.com/chapter/10.1007/978-1-4757-3510-9_1?error=cookies_not_supported&code=38aed46c-ca9a-4204-aeca-cef7d85eb076