

UNIVERSIDAD DE GUADALAJARA  
Centro Universitario de Ciencias Exactas e Ingenierías  
Departamento de Ciencias Computacionales

6<sup>to</sup> semestre

Tema: Kbhit



Materia:

Seminario de Sistemas Operativos

Docente:

Violeta del Rocío Becerra Velázquez

PRESENTA:

Daniel Martínez Martínez

Código:

217565958

Carrera:

Ingeniería en Computación.

Sección:

D02

Fecha de entrega:

Domingo, 12 de Febrero de 2023

## Índice

¿Qué significa JCL?	2
¿Cuál es la diferencia entre el procesamiento por lotes sencillo y el multiprogramado?	2
Evolución sistemas operativos	2
Int86 en C	3
¿Para qué sirve Kbhit?	3
Equivalentes de Kbhit en otros lenguajes	3
¿Qué lenguaje de programación es el que utiliza y por qué?	3
Conclusiones	3
Bibliografía	4

### ¿Qué significa JCL?

Job Control Language (JCL) es un conjunto de instrucciones que se codifican para decirle a los sistemas operativos mainframe de IBM acerca del trabajo que se desea ejecutar. JCL proporciona un medio de comunicación entre el programa de aplicación, el sistema operativo y el hardware del sistema.

Todos los *Jobs* usan tres tipos principales de sentencias:

- Una sentencia JOB para identificar la unidad de trabajo que el sistema operativo va a ejecutar.
- Una o más sentencias EXEC, usadas para marcar el inicio de un paso dentro de un Job, además especifica el nombre de un programa o procedimiento catalogado para ser ejecutado.
- Una o varias sentencias DD para identificar los conjuntos de información de entrada/salida usados.

### ¿Cuál es la diferencia entre el procesamiento por lotes sencillo y el multiprogramado?

El procesamiento por lotes es aquel donde se atienden grandes números de tareas, siendo ejecutadas una por una por el sistema, sin necesidad de que el usuario intervenga.

Por otro lado, el procesamiento por lotes multiprogramado es aquel donde el sistema operativo puede ejecutar múltiples procesos. Se ejecutan múltiples programas a la vez para evitar que el CPU y la memoria queden sin usar.

En el procesamiento por lotes, los Jobs tienen que ser ejecutados uno tras otro por la computadora sin intervención del usuario, mientras que en el sistema operativo multiprogramado se pueden ejecutar varias tareas a la vez usando un solo procesador.

### Evolución sistemas operativos

Siguiendo la evolución de los sistemas operativos, ¿Qué fue lo que le siguió al procesamiento por lotes con multiprogramación?

- Los sistemas con procesamiento por lotes multiprogramados son considerados la **tercera generación** de los sistemas operativos (1965-1980).
- Cuarta generación (1980-Actualidad): Son los sistemas operativos con los que contamos actualmente en todos nuestros dispositivos personales. Algunas de sus características más destacadas son...
  - Se encuentran a precios accesibles en múltiples dispositivos del mercado.
  - Cuentan con interfaces gráficas amigables para facilitar el uso del dispositivo por parte del usuario.
  - Permiten la interconexión de todos los usuarios, creando una gran red de computadoras por todo el mundo. El origen de las grandes redes trajo consigo la creación de los sistemas operativos de red y los sistemas distribuidos.
    - Los sistemas operativos de red son sistemas operativos diseñados para soportar estaciones de trabajo, computadoras personales y otros dispositivos que estén conectados a una red de área local. Este software permite que varios dispositivos de una red se comuniquen y compartan recursos entre sí.

- Un sistema distribuido es un sistema de software cuyos componentes están separados físicamente y conectados entre sí por una red de computadoras, se comunican y coordinan entre ellos pasando mensajes. Dichos componentes interactúan entre ellos para lograr una meta común.

### Int86 en C

Int86 es una función de librería de C la cual facilita el acceso a las interrupciones de los servicios de DOS y de la BIOS. Se declara de la forma:

***int86(int intnum, union REGS \*in, union REGS \*out)***

El número de la interrupción en esta función es intnum, in es una unión que contiene los registros que se usarán para pasar la información a los manejadores de la interrupción y out es una unión que guardará los valores devueltos por la interrupción (si los hay).

- Puede ser usada para acceder a los movimientos del cursor, entradas de teclado, modos de video, etc.

### ¿Para qué sirve Kbhit?

La función \_kbhit comprueba en la consola si se ha presionado una tecla recientemente. Si la función devuelve un valor distinto de cero, hay una pulsación de tecla esperando en el búfer. A continuación, el programa puede llamar a \_getch o \_getche para obtener la pulsación de tecla.

Es una función listener que siempre está al pendiente de si se realiza una pulsación en el teclado.

### Equivalentes de Kbhit en otros lenguajes

- En C# existe algo parecido, llamado KeyPress
- En java se puede usar la interface KeyListener
- Python: Se puede usar Kbhit a través del módulo msvcrt -> ***msvcrt.kbhit()***

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

Usamos Python porque contiene muchas herramientas muy útiles para el desarrollo de algoritmos complejos de una forma medianamente simplificada. El uso de las funciones que ofrece el lenguaje hacen que se reduzca en gran medida el tamaño de los segmentos de código, lo que mejora mucho su legibilidad y capacidad de mantenimiento.

### Conclusiones

Los sistemas operativos evolucionaron para eficientar la realización de las tareas de distintas formas, tales como el aprovechamiento al máximo de los recursos de procesamiento de un solo ordenador para tenerlo haciendo alguna tarea en todo momento o la implementación de interconexiones en red entre dispositivos y sistemas para la solución de objetivos en común.

Con respecto a las funciones como Kbhit para captar eventos en el teclado, considero que son indispensables en los programas de hoy en día, ya que muchas veces se implementan sistemas cuyo objetivo principal es el uso cómodo y rápido de los dispositivos de entrada y salida. Sin estas funciones, estaríamos sometidos a estar presionando enter de forma frecuente en muchos programas que existen actualmente.

### Bibliografía

- Tanenbaum, A. S. (2008) Modern Operating Systems Pearson Educación, México.
- *Basic JCL concepts*. (s. f.). © Copyright IBM Corporation 2010. <https://www.ibm.com/docs/en/zos-basic-skills?topic=collection-basic-jcl-concepts>
- *Difference between Batch Operating System and Multiprogramming Operating System - javatpoint*. (s. f.). [www.javatpoint.com](https://www.javatpoint.com/batch-operating-system-vs-multiprogramming-operating-system). <https://www.javatpoint.com/batch-operating-system-vs-multiprogramming-operating-system>
- GeeksforGeeks. (2022, 29 marzo). *Difference between Batch Processing OS and Multiprogramming OS*. <https://www.geeksforgeeks.org/difference-between-batch-processing-os-and-multiprogramming-os/>
- T. (2022, 2 diciembre). *\_kbhit*. Microsoft Learn. <https://learn.microsoft.com/es-es/cpp/c-runtime-library/reference/kbhit?view=msvc-170>
- *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>