

Seminario de Sistemas Operativos



Departamento de Ciencias Computacionales
Centro Universitario de Ciencias Exactas e Ingenierías

Universidad de Guadalajara

Violeta del Rocío Becerra Velázquez

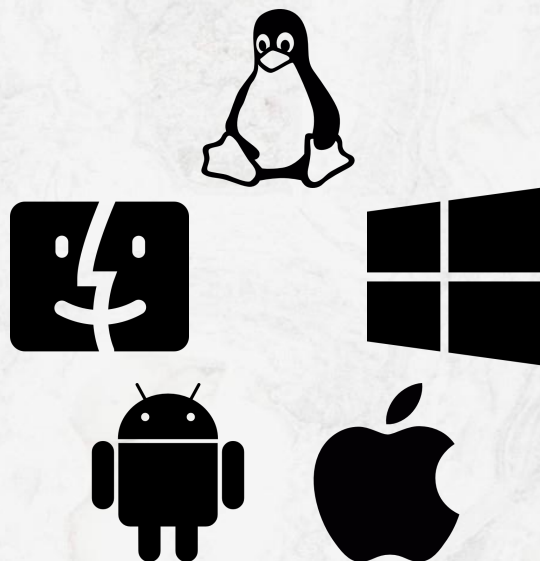
Saul Alejandro Castañeda Pérez

Actividad de Aprendizaje 3

13:00 - 14:55

217564323

D02



Índice

JCL	3
Procesamiento por lotes y multiprogramación	3
Evolución de los SO	4
Int86 en C	6
Función Kbhit	6
Equivalentes de Kbhit	6
Elección de lenguaje de programación	7
Conclusión	7
Referencias	7

JCL

Job Control Language (JCL) es un lenguaje de programación que permite codificar las instrucciones necesarias para la ejecución de un proceso batch, estas instrucciones o sentencias son interpretadas y ejecutadas por el Gestor de trabajos (JES).

Sentencias JOB / EXEC / DD

- **JOB:** Identifica el trabajo a realizar. Es procesada por la parte de lectura del JES(Reader). Aporta un registro contable para el trabajo.
- **EXEC:** Identifica el programa o procedimiento que se va a ejecutar. Se procesa por parte del JES en el apartado (Converter) destinado a modificar el JCL por rutinas y sentencias estándar del sistema . Se abre un registro contable que identifica el paso de trabajo.
- **DD:** Identifica los recursos que va a utilizar el programa (por ejemplo, ficheros). Como la sentencia anterior la ejecución de la sentencia es procesada en primer lugar por el apartado (Converter) del JES. En algunos casos si ello fuese preciso se acude al Gestor de Recursos del Sistema (SRM).

Procesamiento por lotes y multiprogramación

Procesamiento por lotes: una serie de trabajos se ejecutan sin intervención humana en el sistema de procesamiento por lotes. En este conjunto de trabajos con necesidades similares se agrupan en lotes y se ingresan a la computadora para su ejecución.

Multiprogramación: el sistema operativo de multiprogramación permite ejecutar múltiples procesos al monitorear sus estados de proceso y cambiar entre procesos. Ejecuta múltiples programas para evitar la infrautilización de la CPU y la memoria.

Diferencias	
Procesamiento por lotes	Multiprogramación
El sistema de procesamiento por lotes es más lento que el sistema de multiprogramación.	El sistema de multiprogramación es más rápido en el procesamiento que el sistema de procesamiento por lotes.
En el sistema de procesamiento por lotes, la CPU debe permanecer inactiva.	En el sistema de multiprogramación, la CPU no necesita permanecer inactiva.
En el procesamiento por lotes, la ejecución del sistema operativo del proceso comienza por lotes.	En la programación múltiple, la ejecución del proceso comienza cuando centra el procesador.

En el procesamiento por lotes, agrupación de varios trabajos de procesamiento para ser ejecutados uno tras otro por una computadora sin ninguna interacción del usuario.	Capacidad del sistema operativo de programación múltiple de un sistema operativo para ejecutar varios programas al mismo tiempo en un solo procesador.
En el sistema de procesamiento por lotes, la utilización de la CPU es menor.	En el sistema de multiprogramación, la utilización de la CPU es mayor.

Evolución de los SO

- Años 40

A finales de los años 1940, con lo que podríamos llamar la aparición de la primera generación de computadoras, se accedía directamente a la consola de la computadora desde la cual se actuaba sobre una serie de micro interruptores que permitían introducir directamente el programa en la memoria de la computadora.

- Años 50

A principios de los años 50 con el objeto de facilitar la interacción entre persona y computador, los sistemas operativos hacen una aparición discreta y bastante simple, con conceptos tales como el monitor residente, el proceso por lotes y el almacenamiento temporal.

- Monitor residente

Su funcionamiento era bastante simple, se limitaba a cargar los programas a memoria, leyéndolos de una cinta o de tarjetas perforadas, y ejecutarlos, el problema era encontrar una forma de optimizar el tiempo entre la retirada de un trabajo y el montaje del siguiente.

- Procesamiento por lotes

Como solución para optimizar el tiempo de montaje surgió la idea de agrupar los trabajos en lotes, en una misma cinta o conjunto de tarjetas, de forma que se ejecutan uno a continuación de otro sin perder apenas tiempo en la transición.

- Almacenamiento temporal

Su objetivo era disminuir el tiempo de carga de los programas, haciendo simultánea la carga del programa o la salida de datos con la ejecución de la siguiente tarea. Para ello se utilizaban dos técnicas, el buffering y el spooling.

- Años 60

En los años 1960 se produjeron cambios notorios en varios campos de la informática, con la aparición del circuito cerrado la mayoría orientados a seguir incrementando el potencial de los computadores, para ello se utilizaban técnicas de lo más diversas:

- **Multiprogramación**

En un sistema multiprogramado la memoria principal alberga a más de un programa de usuario, la CPU ejecuta instrucciones de un programa, cuando el que se encuentra en ejecución realiza una operación de E/S; en lugar de esperar a que termine la operación de E/S, se pasa a ejecutar otro programa. Si éste realiza, a su vez, otra operación de E/S, se mandan las órdenes oportunas al controlador, y pasa a ejecutarse otro.

De esta forma es posible, teniendo almacenado un conjunto adecuado de tareas en cada momento, utilizar de manera óptima los recursos disponibles.

- Tiempo compartido

En este punto tenemos un sistema que hace buen uso de la electrónica disponible, pero adolece de falta de interactividad; para conseguirla debe convertirse en un sistema multiusuario, en el cual existen varios usuarios con un terminal en línea, utilizando el modo de operación de tiempo compartido.

En estos sistemas los programas de los distintos usuarios residen en la memoria, al realizar una operación de E/S los programas ceden la CPU a otro programa, al igual que en la multiprogramación, pero a diferencia de ésta, cuando un programa lleva cierto tiempo ejecutándose el sistema operativo lo detiene para que se ejecute otra aplicación, con esto se consigue repartir la CPU por igual entre los programas de los distintos usuarios, y los programas de los usuarios no se sienten demasiado lentos por el hecho de que los recursos sean compartidos y aparentemente se ejecutan de manera concurrente.

- Tiempo real

Estos sistemas se usan en entornos donde se deben aceptar y procesar en tiempos muy breves un gran número de sucesos, en su mayoría externos al ordenador, si el sistema no respeta las restricciones de tiempo en las que las operaciones deben entregar su resultado se dice que ha fallado, el tiempo de respuesta a su vez debe servir para resolver el problema o hecho planteado.

El procesamiento de archivos se hace de una forma continua, pues se procesa el archivo antes de que entre el siguiente, sus primeros usos fueron y siguen siendo en telecomunicaciones.

- Multiprocesador

Permite trabajar con máquinas que poseen más de un microprocesador, en un multiprocesador los procesadores comparten memoria y reloj.

- Sistemas operativos desarrollados

Además del Atlas Supervisor y el OS/360, los sesenta marcaron el inicio de UNIX, a mediados de los 60 aparece Multics, sistema operativo multiusuario – multitarea desarrollado por los laboratorios Bell de AT&T y programado en PL/1 uno de los pocos SO desarrollados en un lenguaje de alto nivel en aquel tiempo, luego del fracaso del proyecto UNIX comienza a desarrollarse a partir de este a finales de la década.

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

Función 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, 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 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()`

Elección de lenguaje de programación

Para realizar las prácticas de esta materia elegimos el lenguaje de programación Python ya que es un lenguaje con el que ya hemos trabajado antes, por lo que ya contamos con cierta experiencia, además de que es muy sencillo realizar interfaces gráficas que se vean decentes sin mucho trabajo, permitiéndonos enfocarnos en las funcionalidades del sistema.

Conclusión

Los sistemas operativos han ido evolucionando a medida de las necesidades que se fueron generando, cada sistema operativo tiene un fin determinado que es la de realizar tareas según el objetivo a lograr, dependiendo de lo que necesite el o los usuarios, la mayoría de los sistemas operativos de última generación tienden a atender un gran número de usuarios y que los procesos a realizar demoren un mínimo de tiempo.

Referencias

colaboradores de Wikipedia. (2022b, diciembre 18). Job Control Language. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Job_Control_Language

Contributor, T. (2021, 9 julio). JCL (job control language). Data Center. <https://www.techtarget.com/searchdatacenter/definition/JCL>

Greyrat, R. (2022, 5 julio). Diferencia entre el sistema operativo de procesamiento por lotes y el sistema operativo multiprogramación – Barcelona Geeks. <https://barcelonageeks.com/diferencia-entre-el-sistema-operativo-de-procesamiento-por-lotes-y-el-sistema-operativo-de-multiprogramacion/>

Evolución de los S.O. (s. f.). <https://users.dcc.uchile.cl/%7Elmateu/CC41B/Apuntes/evolucion/historia.htm>

TP4 Evolucion SO - AlvaroMangione. (s. f.). <https://sites.google.com/site/alvaromangione/home/5to-info/software-iii/tp4-evolucion-so>

USO DE LOS RECURSOS DE UN EQUIPO DE CÓMPUTO MEDIANTE C++. (s. f.). itnuevolaredo. Recuperado 5 de febrero de 2023, de <http://www.itnuevolaredo.edu.mx/takeyas/Articulos/ProgramacionC/ARTICULO%20Uso%20de%20los%20recursos%20de%20un%20equipo%20de%20computo%20mediante%20C.pdf>

T. (2022b, diciembre 2). _kbhit. Microsoft Learn. Recuperado 5 de febrero de 2023, de <https://learn.microsoft.com/es-es/cpp/c-runtime-library/reference/kbhit?view=msvc-170>