

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO
SISTEMAS OPERATIVOS

TAREA 02

Debianers

Integrantes

Yosshua Cisneros - 179889

Rodrigo Plauchu - 182671

Mauricio Gutiérrez - 183014

Fecha (s) de elaboración de la tarea

17 de Febrero de 2022

Tarea AE

SOs con diferentes Características Funcionales

Analice usted, cada una de las siguientes ocho combinaciones (de las tres características funcionales) que darían origen a ocho posibles Sistemas Operativos Genéricos:

1. Mono-programación; No Tiempo Compartido; Monousuario.

El CPU sólo atiende un proceso a la vez y los procesos sólo pueden ser de un solo usuario, en determinado tiempo. En la memoria sólo existe un proceso y cada nuevo proceso se inicia cuando el anterior haya terminado (no hay time slice). Por lo tanto si se inicia una I/O en el proceso, el CPU no puede ejecutar otro proceso porque sólo existe uno en memoria.

Por las características anteriores, los tiempos de ejecución serían muy largos para cada proceso y usuario, en determinado tiempo.

2. Mono-programación; No Tiempo Compartido; Multiusuario.

A comparación del SO anterior, en este se pueden ejecutar procesos de usuarios diferentes; sin embargo, el CPU sigue almacenando un proceso en la memoria y comienza a ejecutar el siguiente proceso hasta que el anterior haya terminado (tampoco existe time slice). En comparación con el SO anterior, este es más justo porque los recursos se van alternando entre los diferentes usuarios (el usuario en determinado tiempo no tiene que esperar a que los demás usuarios terminen sus procesos).

3. Mono-programación; Tiempo Compartido; Monousuario.

El CPU sigue teniendo un proceso en la memoria, pero todos los procesos son del mismo usuario. Debido a que sólo hay un proceso en la memoria, el time slice podría alargar (o no) los tiempos de ejecución y el CPU no se ocuparía durante cada time slice o hasta que se inicie una I/O. Un nuevo proceso podrá ser ejecutado hasta que el anterior haya terminado. Dependiendo del tamaño del time slice, los tiempos de ejecución podrían ser más equitativos entre los nuevos procesos, esto a comparación de los SO anteriores.

4. Mono-programación; Tiempo Compartido; Multiusuario.

El CPU sigue teniendo un proceso en la memoria, pero los procesos pueden ser de diferentes usuarios. Si bien se consigue que el SO pueda ejecutar procesos de otros usuarios, se siguen teniendo las mismas características del SO anterior, es decir, un sólo proceso en memoria y un time slice. Todo dependerá del tamaño del time slice para que el CPU funcione en gran parte del tiempo o no. Sin embargo, el nuevo proceso sólo puede iniciarse cuando el último proceso haya terminado. Además, en comparación con el anterior SO, el uso de los recursos en este SO es más equitativo entre usuarios, ya que el proceso de un usuario no puede monopolizar los recursos del SO.

5. Multiprogramación; No Tiempo Compartido; Monousuario.

A comparación de los modelos anteriores, en este SO el CPU puede almacenar varios procesos en la memoria, es decir permite ejecutar otros procesos cuando se inicia una I/O o cuando termina el proceso anterior. Sin embargo, este SO continúa dedicando todos los recursos del CPU a un sólo proceso del único usuario. Por lo tanto, los tiempos de ejecución no serán equitativos entre procesos (no hay time slice). De las ventajas destacables en comparación con los SO anteriores es el hecho de mantener otros procesos en la memoria, ya que el CPU continúa en uso con otros procesos cuando se inicia una I/O o cuando

termina el proceso anterior del mismo usuario. A comparación con los SO anteriores, este SO permite maximizar la utilización del CPU y ejecutar más procesos en menor tiempo.

6. Multiprogramación; No Tiempo Compartido; Multiusuario.

SO1 o SO2

La multiprogramación se apoya en varios elementos del hardware: la interrupción, el DMA y el canal. En un sistema multiprogramado la memoria principal alberga a más de un programa de usuario. La CPU ejecuta instrucciones de un programa, cuando el programa en ejecución (es decir, el que ocupa la CPU) realiza una operación de I/O, emite ciertas órdenes al controlador (al igual que en los sistemas monoprogramados); pero en lugar de esperar a que termine la operación de I/O comprobando el bit de ocupación, se pasa a ejecutar otro programa. Si este nuevo programa realiza, a su vez, otra operación de I/O, se mandan las órdenes oportunas al controlador, y pasa a ejecutarse otro programa. Esto permite que varios dispositivos trabajen simultáneamente, además, en la CPU no se tienen que ejecutar ciclos de comprobación del estado de los dispositivos. Varios usuarios interactivos. No hay interacción + lotes. Como ventajas mantiene más de 1 proceso en memoria, y que se proporciona un DMA a varios dispositivos, lo cual conviene porque resulta más económico que tener un controlador DMA por dispositivo.

7. Multiprogramación; Tiempo Compartido; Monousuario.

A diferencia del anterior este tipo de sistema sigue brindado soporte solo a un usuario, pero le permite realizar varias tareas a la misma vez, CPU almacena varios procesos en memoria. Al ser capaz de ejecutar diferentes procesos en un mismo momento, se pueden ejecutar diferentes programas y aplicaciones. Aún así, para que el CPU pueda transferirse al siguiente proceso es necesario que el proceso anterior inicie una I/O, que el proceso anterior termine o que el time slice se termine. Esto, por supuesto, aumenta mucho la productividad del sistema y suple esta necesidad creada por los sistemas mono-programados.

8. Multiprogramación; Tiempo Compartido; Multiusuario.

SO3

En el tiempo compartido los programas de los distintos usuarios residen en memoria; al realizar una operación de I/O los programas ceden la CPU a otro programa, al igual que en la multiprogramación. Pero, a diferencia de ésta, el CPU no espera a que termine el proceso o que haya I/O, sino que puede detener el proceso si este ha terminado su tiempo asignado (time slice). Con esto se consigue repartir la CPU con equidad entre los programas de los distintos usuarios. Al tener la CPU una capacidad de cálculo tan grande, los programas de los usuarios no se sienten demasiado ralentizados por el hecho de que la CPU (y los demás recursos) sean compartidos. Por concurrencia se entiende la existencia de varias actividades simultáneas o paralelas.

Las desventajas podrían ser que hay un mayor uso de procesador, de memoria, de disco, de impresora, sin embargo hay un menor tiempo transcurrido entre procesos, mayor tasa de productividad y menor tiempo de respuesta. Por lo que podríamos decir que utiliza más recursos y podría ser desventajoso para ciertos propósitos.