



Programación de servicios y procesos

Act1.1. Introducción a los
sistemas multitarea

Francisco José García Cutillas | 2FPGS_DAM



Índice

Ejercicio 1 3

Ejercicio 2 3

Ejercicio 3 3

Ejercicio 4 4

Ejercicio 1

Explica las diferencias entre los lenguajes de programación interpretados y compilados. Escribe un par de ejemplos de cada uno.

En un lenguaje compilado la máquina de destino es la que tiene que ser capaz de traducir el programa. Sin embargo, en un lenguaje interpretado, es el intérprete el encargado de traducir el programa a un lenguaje que la máquina sea capaz de comprender.

Como lenguajes compilados podemos tener a C, C++ o Java. Mientras que como lenguajes interpretados tenemos PHP, JavaScript o Python.

Ejercicio 2

Describe las diferencias entre programa y proceso. Pon un ejemplo comparado con la vida real.

Un programa es una secuencia de instrucciones escrita en un lenguaje dado, mientras que un proceso es la ejecución de dicho programa.

Como ejemplo podemos tener el Microsoft Word. El programa está disponible en la máquina, pero hasta que no se ejecuta no iniciamos el proceso en sí de este programa.

Ejercicio 3

Explica en qué consiste la programación distribuida.

La programación distribuida consiste en redes de ordenadores conectados entre sí que colaboran para resolver complejos cálculos basándose en la división de tareas, con el objetivo de optimizar la gestión de dichas tareas y encontrar soluciones flexibles.

Ejercicio 4

Encuentra las diferencias entre ejecutar dos procesos o realizar una bifurcación o fork.

Explica razonadamente dichas diferencias, si puedes pon un ejemplo de cada uno.

La diferencia es que al ejecutar dos procesos, tendríamos dos procesos distintos. Mientras que al realizar una bifurcación el proceso seguiría siendo el mismo. El proceso original (padre) produciría una copia idéntica de él (hijo). La copia continúa con el estado del padre, aunque cada uno mantendría su propio estado de memoria.

Un ejemplo de procesos independientes podría ser la ejecución de dos programas distintos como pudieran ser Chrome y Microsoft Word.

Un ejemplo de fork podría ser un servidor web en el que se produce una bifurcación para crear un nuevo proceso hijo cada vez que se atiende una solicitud de cliente. Esto permite que el servidor maneje múltiples solicitudes de manera concurrente sin interrumpir otras operaciones.