

Universidad Nacional Autónoma de México
Facultad de Ciencias
Sistemas Operativos

Bernal Martínez Fernando
Luna Vázquez Felipe Alberto
Hurtado Gutiérrez Marco Antonio
Hernández Chávez Jorge Argenis

Febrero 2019



1 Preguntas de la Practica 01

1.1 ¿Por qué en Pintos los procesos de kernel se llaman threads en lugar de procesos?

Un thread o un thread de control es una serie de instrucciones (executable steps). Un thread de control es un programa en ejecución. Cada thread posee todo lo necesario para su ejecución: una pila, código y segmentos de datos. Y de acuerdo con Silberschatz tenemos la siguiente definición, "Un proceso generalmente incluye la pila de ejecución (process stack), la cual contiene datos temporales". En el caso de Pintos el Bloque de Control de Proceso (PCB) se encuentra definido en el archivo `src/threads/thread.h`

1.2 ¿Para qué sirve el stack de un proceso?

En este caso el PCB de Pintos hace referencia explícita de los datos locales y lo hace mediante el apuntador `stack`, que hace referencia a la pila de ejecución y así almacenar las variables locales y los saltos.

1.3 En Pintos cada thread utiliza un bloque de 4KB, en el cual se almacena su PCB y el stack del thread. ¿Dónde se almacenan las variables globales?

En el tiempo de ejecución de C, las variables globales se almacenan en uno de dos lugares; el segmento de datos o el segmento BSS. La forma en que determina a cuál pertenece una variable en particular es si se inicializa o no.

1.4 Si suponemos que solamente existe un thread en ejecución y dicho thread se bloquea (utilizando la función `thread_block`). ¿De que forma se podrá despertar el thread si es el único en ejecución?

Recordemos que hay un thread que nunca se duerme que siempre esta activo y este será el encargado de despertarlo, en pintos necesariamente se debe hacer mediante la llamada a función `thread_unblock()`.