

### 1) Sincronización

Suponga que debe implementar un sistema operativo desde cero para correr exclusivamente en un procesador de muy pequeña capacidad que va a estar conectado a varios instrumentos de meteorología. Distintos procesos samplearán los diversos sensores y colocarán en memoria compartida las distintas mediciones. A la vez, otros procesos tomarán esa información, realizarán cálculos y volcarán el resultado también en memoria compartida. Todo esto requiere el uso de primitivas de sincronización.

¿Qué primitivas brindaría y por qué? Justifique basando sus argumentos tanto en las características generales de las primitivas como así también en el escenario específico que plantea este ejercicio.

### 2) Procesos

- Describa la PCB, indicando modificaciones necesarias para soportar threads.
- Proponga un escenario en donde un proceso requiera la modificación de algún valor de la PCB, y escriba el pseudocódigo para realizar ese cambio.

### 3) Administración de entrada/salida

¿Cuáles son las 2 grandes diferencias, a nivel de la administración de entrada/salida que hace el sistema operativo, entre los discos magnéticos (HDD) y los "discos" de estado sólido (SSD)?

### 4) Sistemas de archivos

Se quiere implementar el ZapatoFileSystem, que es un sistema de archivos que incorpora un atributo que indica si un archivo contiene la palabra "zapato". Para hacerlo se desea modificar el código de la syscall write() de manera tal que al detectar la palabra "zapato" dentro del buffer que se le pasa a write() se prenda un atributo especial en ese archivo.

Describa qué sistema de archivos tomaría como base para agregarle esa funcionalidad (FAT o inodos) y por qué, tratando de minimizar el impacto de performance o la facilidad de implementación (a su gusto).

Describa esquemáticamente la implementación.

### 5) Seguridad

Una función ejecuta cierto código protegido sólo cuando se le suministran ciertos nombres de usuario.

Describir al menos dos maneras en que un atacante con suficiente tiempo y capacidad de invocarla con los parámetros que desee puede lograr la ejecución del código protegido.

```
void f(char *usuario) {
    char copia_usuario[14];
    int puede_pasar = chequear_nivel_permisos(usuario);
    strcpy(copia_usuario, usuario);
    if (puede_pasar) {
        // Ejecuta código protegido.
    }
}
```

