

Sistemas Operativos

Departamento de Computación – FCEyN – UBA
Primer cuatrimestre de 2024

Segundo parcial – Primer cuatrimestre de 2024

1	2	3	4	Nota
B	B	B	B	A

- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas.
- Entregue esta hoja junto al examen, la misma **no** se incluye en la cantidad total de hojas entregadas.
- Cada ejercicio debe realizarse en hojas separadas y numeradas. Debe identificarse cada hoja con nombre, apellido, LU y número de orden.
- Cada código o pseudocódigo debe estar bien explicado y justificado en castellano. ¡Obligatorio!
- Toda suposición o decisión que tome deberá justificarla adecuadamente. Si la misma no es correcta o no se condice con el enunciado no será tomada como válida y será corregida acorde.
- La devolución de los exámenes corregidos es personal. Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- Cada ejercicio se calificará con Bien, Regular o Mal, dependiendo del apego de la resolución al punto central siendo evaluado.
- Los parciales se aprueban con al menos dos ejercicios B y uno R. En otro caso, están desaprobados. No llevan nota numérica.

Ejercicio 1.

Para un *filesystem* basado en inodos, responda las siguientes preguntas/consignas, justificando:

- Escribir el pseudo código para acceder al archivo A/B/c.txt, considerando que se tiene el inodo del directorio A.
- ¿Cuántas escrituras a disco requiere renombrar el archivo /home/juan/listado.txt a /home/maria/archivo.txt?
- Suponiendo que puede ejecutar un programa como root, ¿podría crear un loop infinito al leer un archivo? En caso de respuesta afirmativo, explique cómo podría producirse. En caso contrario, justifique.

Ejercicio 2.

Se crea un programa para que los usuarios puedan modificar sus contraseñas con el siguiente código:

```
int main(void) {
    cambiarPassword();
    return 0;
}

void cambiarPassword() {
    int numeroDeUsuario = obtenerUsuarioActual();

    char password[250];
    char passwordConfirmacion[250];
    printf("Ingrese su password actual");
    fgets(password, 250, stdin);

    if (hash(password) == hashDePasswordActual(numeroDeUsuario)) {
        printf("Ingrese su nueva password");
        fgets(password, 250, stdin);
        printf("Confirme su nueva password");
        gets(passwordConfirmacion);

        if (sonIguales(password, passwordConfirmacion)) {
            actualizarPassword(numeroDeUsuario, password);
        }
    }
}
```

- a) Explique qué problema de seguridad tiene el código mencionado, y cómo debería explotarlo un atacante.
- b) ¿Qué podría hacer con las contraseñas un atacante que logra explotar la vulnerabilidad?

Ejercicio 3.

Considere los siguientes elementos:

- i) La pantalla táctil de un celular con Android.
- ii) Un disco virtual, que permite a los usuarios guardar archivos y sincronizarlos en una carpeta remota mediante la red (al estilo Dropbox, Google Drive).
- iii) Un GPU (Unidad de Procesamiento Gráfico) que se utiliza para multiplicaciones de matrices de cálculos científicos.

En base a ellos, responda y justifique las siguientes preguntas, para cada caso:

- a) ¿Tiene sentido que el driver use un buffer para almacenar una mayor cantidad de entrada del usuario o la salida antes de enviarla / recibirla del dispositivo, o conviene intercambiarla en cuanto esté disponible?
- b) ¿Qué mecanismo de entrada/salida utilizaría?

Ejercicio 4.

Un sistema de subastas online distribuido debe funcionar con las siguientes características:

- i) Los adquirientes son procesos que funcionan en diversos nodos de una red distribuida.
- ii) El subastador es un proceso más, que funciona en otro nodo de la red.
- iii) Cada ronda de la subasta funciona en dos etapas: el subastador oferta el lote correspondiente indicando el tipo de bien y el precio unitario, y lo asigna al primer oferente que se manifiesta interesado, ahí se informa la cantidad de elementos que contiene el lote, y se espera confirmación o negación de parte del oferente.
- iv) Es requisito indispensable asignar cada lote subastado al proceso que oferte primero, en sentido temporal.
- v) Una vez asignado el lote, se debe confirmar la operación: el subastador indica de cuántos elementos se compone el lote (eg, "lote compuesto por 20 unidades"), y el adquiriente debe confirmar si quiere todo el lote o desiste de la operación. No se admiten compras parciales.

Para implementar este sistema se propone que las ofertas se realicen mediante el envío de un mensaje al nodo subastador, y que la confirmación de la operación se realice mediante el siguiente protocolo: cada oferente tiene un directorio específico para esta aplicación en su propio disco rígido, y el subastador crea un archivo llamado *ganasteS* en el correspondiente al ganador de la oferta, conteniendo el tamaño del lote. El ganador responde creando un archivo llamado *respuesta* que contiene la palabra sí o la palabra no para indicar si lo acepta o no. El acceso a los discos remotos se hace mediante una aplicación distribuida cuyo cliente está en el subastador y sus servidores en los adquirientes. El cliente envía mensajes al servidor correspondiente pidiendo lectura o escritura de tal o cuál archivo.

Responda justificando:

- a) ¿El protocolo propuesto permite cumplir con el requisito iv)? *no*
- b) ¿El protocolo propuesto para resolver el requisito v), funciona? Si no, ¿qué problemas tiene? Si sí, ¿bajo qué suposiciones? *no se pueden manejar mensajes (?)*
- c) Qué algoritmos de los analizados en la materia podrían servir para realizar implementaciones más robustas de este protocolo. *2PC*