

SISTEMAS DISTRIBUIDOS

Primer Cuatrimestre 2020

Laboratorio N° 2

Todos los códigos fuente deberán subirse de forma independiente y luego se deberá subir el archivo con las respuestas.

Registro:	Apellido y Nombre:
Registro:	Apellido y Nombre:

Ejercicios a Resolver

1. Si el paradigma de comunicación es asincrónico, ¿es también persistente? Explique su respuesta y brinde un ejemplo.
2. Dos divisiones de la armada de Rocolandia, Rojos y Azules, están acampados en 2 colinas cercanas. En el valle que está entre las 2 colinas está ubicada la división enemiga Zapatilandia. Las divisiones de Rocolandia están seguras donde están ubicados sus campamentos, y pueden intercambiar confiadamente mensajes a través del valle. ¿Pueden llegar a un acuerdo para atacar al enemigo en forma conjunta si el modelo de sistema es asincrónico? Justifique.
3. Considere un cliente de correo electrónico y conteste las siguientes preguntas. Explique las respuestas.
 - a) ¿Cuál modelo de comunicación le parece el más apropiado?
 - b) ¿Utilizaría algún nivel de sincronismo?
4. Realizar un experimento para obtener el tiempo de ejecución en la escritura de un pipe y en la escritura de un archivo. El nombre del archivo fuente debe comenzar con la palabra *escribir*. Explique brevemente los resultados obtenidos. La precisión de las mediciones tienen que estar en **microsegundos**.
5. Realizar un experimento para obtener el tiempo de ejecución en la creación de una cola de mensajes y la escritura de un mensaje. El nombre del archivo fuente debe *colamen*. Explique brevemente los resultados obtenidos. La precisión de las mediciones tienen que estar en nanosegundos.
6. ¿Qué es el binder? Enumere las formas de ubicar un binder. Explique la sobrecarga que se produce en la comunicación para cada uno de los casos.

7. Considere la información de la figura 1

a) ¿Qué representa el código de la figura 1? Explique cada uno de los elementos.

```
#define TAMANO_BLOQUE 512
struct stat_archivo
{
    int hubo_error;
    char descripcion_error[255];
    int st_mode; /* protección */
    int st_nlink; /* nro de enlaces físicos */
    int st_uid; /* ID del usuario propietario */
    int st_gid; /* ID del grupo propietario */
    long int st_size; /* tamaño total en bytes */
    unsigned long st_blksize; /* tamaño de bloque
para el sistema de archivo E/S */
    unsigned long st_blocks; /* nro de bloques asig */
};
struct archivo_escritura
{
    char nombre[512];
    long desplazamiento;
    int datos[TAMANO_BLOQUE];
    int datos_size;
    int hubo_error;
    char descripcion_error[255];
};
struct archivo_lectura
{
    char nombre[512];
    long desplazamiento;
    int datos_size;
};
struct archivo_atributos
{
    char nombre[512];
    int modo;
};

struct camino_t
{
    char nombre[512];
};
typedef struct nodo* lista;
struct nodo
{
    char nombre[255];
    lista siguiente;
};
struct entradas_directorio
{
    lista entradas;
    int hubo_error;
    char descripcion_error[255];
};
struct cambiar_nombre
{
    char antiguo[512];
    char nuevo[512];
};
struct cadena
{
    int hubo_error;
    char descripcion_error[255];
    char nombre[512];
};
struct error
{
    int hubo_error;
    char descripcion_error[255];
};

program FILESERVER
{
    version PRIMERA
    {
        error escribir(archivo_escritura archivo) =1;
        archivo_escritura leer(archivo_lectura entrada) =2;
        stat_archivo obtener_atributos(camino_t camino) =3;
        error modificar_atributos(archivo_atributos archivo) =4;
        error borrar_archivo(camino_t camino) =5;
        error crear_archivo(camino_t camino) =6;
        cadena directorio_actual() =7;
        error cambiar_directorio(camino_t nombre) =8;
        error crear_directorio(archivo_atributos directorio) =9;
        entradas_directorio listar_directorio(camino_t nombre) =10;
        error renombrar(cambiar_nombre nombres) =11;
    } = 1;
}
```

Figura 1: Código

b) ¿Qué se puede generar a partir de esta código?

8. Realizar un cliente que solicite a un servidor las siguientes consultas:

- a) Su tiempo local. El formato que se debe observar es el siguiente: *Thu Oct 5 20:10:27 2011*.
- b) El nombre de la máquina.
- c) La cantidad de usuarios logueados.

Debe estar implementado con RPC.