

s

Instituto Politécnico Nacional

Escuela Superior de Computo

Sistemas operativos (2CM9)

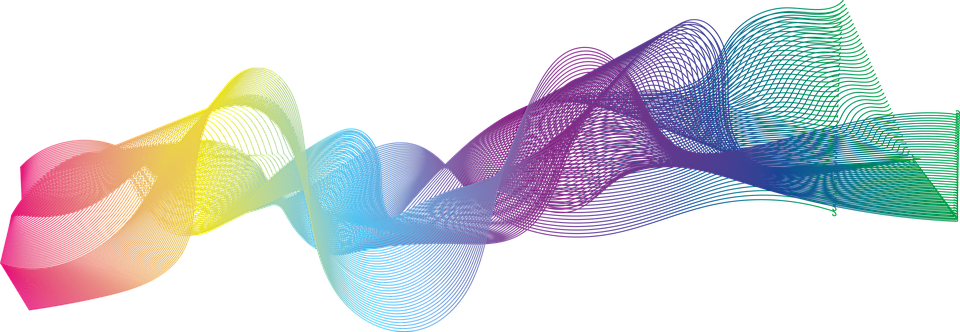
Maestra: Ana Belem Juárez Méndez

**Tarea 3: Ejercicio simulación pseudo-servidor con cola de mensajes**

Alumno:

García González Aarón Antonio

Octubre 02, 2019



Índice

[Codificación 3](#_Toc20926197)

[Capturas de pantalla 6](#_Toc20926198)

[Prueba 1 6](#_Toc20926199)

[Prueba 2 7](#_Toc20926200)

# Codificación

Realizar un programa que simule un servidor, esto es, mediante una terminal simular el comportamiento cliente y desde otra el comportamiento servidor, teniendo las siguientes peticiones con su prioridad correspondiente, donde 1 es la mayor y 4 es la de menor prioridad.

**A continuación, el programa para el proceso cliente:**

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <sys/msg.h>

#include <string.h>

#define CANCELAR 1 // Mayor prioridad

#define RESERVAR 2

#define CONSULTAR 3

#define FINALIZAR 4 // Menor prioridad

struct mensaje{

    long tipo;

    char accion[10]; // Se refiere a cualquier peticion

};

int main(int argc, char const \*argv[]){

    key\_t llave;

    int msgid, tam, control;

    struct mensaje msg;

    llave = ftok("/bin/ls", 10);

    if(llave == -1){

        perror("Error en ftok\n");

        exit(-1);

    }

    msgid = msgget(llave,IPC\_CREAT|0777);

    if(msgid == -1){

        perror("Error en msgget\n");

        exit(-1);

    }

    tam = sizeof(msg) - sizeof(msg.tipo);

    // Enviar mensajes

    control = 0;

    while(control != FINALIZAR){

        printf("1. Cancelar\n2. Reservar\n3. Consultar\n4. Finalizar\n   Seleccione una opcion: ");

        scanf("%d",&control);

        switch(control){

            case CANCELAR:

                strcpy(msg.accion,"CANCELAR");

                msg.tipo=CANCELAR;

                msgsnd(msgid,&msg,tam,0); // Esperara hasta que se pueda enviar el mensaje

                break;

            case RESERVAR:

                strcpy(msg.accion,"RESERVAR");

                msg.tipo=RESERVAR;

                msgsnd(msgid,&msg,tam,0); // Esperara hasta que se pueda enviar el mensaje

                break;

            case CONSULTAR:

                strcpy(msg.accion,"CONSULTAR");

                msg.tipo=CONSULTAR;

                msgsnd(msgid,&msg,tam,0); // Esperara hasta que se pueda enviar el mensaje

                break;

            case FINALIZAR:

                strcpy(msg.accion,"FINALIZAR");

                msg.tipo=FINALIZAR;

                msgsnd(msgid,&msg,tam,0); // Esperara hasta que se pueda enviar el mensaje

                break;

            default:

                printf("ERROR, opcion no valida\n");

                printf("Finalizando ...\n");

                exit(1);

                break;

        }

    }

    printf("Finalizando ...\n");

    return 0;

}

**A continuación, el programa para el proceso servidor:**

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <sys/msg.h>

#include <string.h>

#define CANCELAR 1

#define RESERVAR 2

#define CONSULTAR 3

#define FINALIZAR 4

struct mensaje{

    long tipo;

    char accion[9];

};

int main(int argc, char const \*argv[]){

    key\_t llave;

    int msgid, tam, control;

    struct mensaje msg;

    llave = ftok("/bin/ls", 10);

    if(llave == -1){

        perror("Error en ftok\n");

        exit(-1);

    }

    msgid = msgget(llave,IPC\_CREAT|0777);

    if(msgid == -1){

        perror("Error en msgget\n");

        exit(-1);

    }

    tam = sizeof(msg) - sizeof(msg.tipo);

    control = 0;

    printf("Peticiones acumuladas: \n");

    while(1){

    // Recibir mensajes

        if(msgrcv(msgid,&msg,tam,CANCELAR,IPC\_NOWAIT) == -1){

            if(msgrcv(msgid,&msg,tam,RESERVAR,IPC\_NOWAIT) == -1){

                if(msgrcv(msgid,&msg,tam,CONSULTAR,IPC\_NOWAIT) == -1){

                    if(msgrcv(msgid,&msg,tam,FINALIZAR,IPC\_NOWAIT) == -1){

                        printf("\tFINALIZAR ...\n");

                        break; // Salimos del bucle, a lo mas puede haber un FINALIZAR en toda la ejecucion

                    }

                }

                else{

                    printf("\tCONSULTAR ...\n");

                }

            }else{

                printf("\tRESERVAR ...\n");

            }

        }else{

            printf("\tCANCELAR ...\n");

        }

        control = msg.tipo;

    }

    msgctl(msgid,IPC\_RMID,0); // Liberamos la cola de mensajes

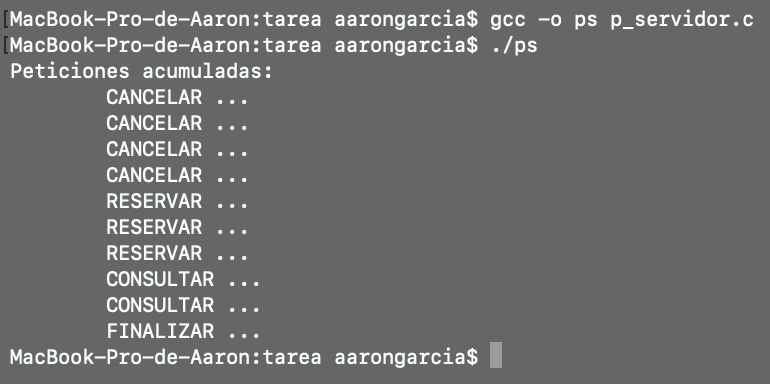
    return 0;

}

# Capturas de pantalla

## Prueba 1





## Prueba 2

