

AMPLIACION DE SISTEMAS OPERATIVOS 15/12/2010

Alumno/a D./Dña.....Especialidad.....

Ejercicio 1.- (4 puntos) Analice los siguientes apartados y **justifique** en su caso las respuestas con ejemplos.

- Concepto de programación en grupo en un SOD. Ponga un ejemplo de la oportunidad de programar en grupo, utilice primitivas de pvm.
- Defina la necesidad de la clase DatagramSocket en el modelo cliente/servidor. Ponga un ejemplo de su utilización.
- Describa como implementar una mejora del algoritmo de Ricart y Agrawala de SCD para que disminuya el paso de mensajes.
- Analice el concepto de habilitaciones. Ponga un ejemplo de su utilización, analizando sus ventajas e inconvenientes

Ejercicio 2 SSL (Secure Sockets Layer) es el protocolo más ampliamente utilizado para implementar servicios criptográficos en aplicaciones web. **(1 punto)** Explicar el código adjunto donde se muestra una comunicación a través de sockets seguros. **(1 punto)** ¿Es un código cliente o servidor? ¿Cuál es la diferencia con sockets inseguros? **(0.5 puntos)** Ventajas e inconvenientes de SSL

```
import java.io.*;
import javax.net.ssl.*;
int port = availablePortNumber;
SSLServerSocket s;
try {
    SSLServerSocketFactory sslSrvFact = (SSLServerSocketFactory) SSLServerSocketFactory.getDefault();
    s = (SSLServerSocket)sslSrvFact.createServerSocket(port);
    SSLSocket c = (SSLSocket)s.accept();
    OutputStream out = c.getOutputStream();
    InputStream in = c.getInputStream();
    catch (IOException e) {
        System.out.println("ERROR : " + e);
    }
}
```

Ejercicio 3.- (1 punto) a) Analice el código indicando cual es su cometido. **(1 punto)** Complete los printf que aparecen en el mismo ¿Esta justificado distribuir este código? **(1.5 puntos)** Indique que misión tienen y de que tipo son las funciones de pvm que aparecen en este código y en el contexto del código

banco.c

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include "pvm3.h"
#define CAJEROS 3
struct persona
{
    int cod;
    float saldo;
};
main()
{
    int i,
        cc,
        tid,
        *idcajeros,
        nprocesos,
        id_buf,
        id_rec,
        cod,
        ok;
    float dinero;
    char op[100];
    char maquina[100];
    int iter;

    FILE *pfich;
    struct persona per;
    struct timeval tmout;
    tmout.tv_sec=25;
    tmout.tv_usec=25;
    tid=pvm_mytid();
    pfich=fopen("datos.dat","wb+");

    if(pfich!=NULL)
    {
        system("clear");
        for(i=1;i<=10;i++)
        {
            printf("\nCliente nº %i", i);
            printf("\n ..");
            scanf("%f", &per.saldo);
            per.cod=i;

            fwrite(&per, sizeof(per), 1, pfich);
        }
        idcajeros=(int *)malloc(CAJEROS*sizeof(int));
        if (idcajeros == NULL)
        {
            printf("\n\n");
            exit(0);
        }
    }
}
```

```

cc = pvm_spawn("cajero", (char**)0, 0, "", CAJEROS, idcajeros);
if (cc != CAJEROS)
{
    printf("\n          \n");
    exit(0);
}
printf("\n          >> %x\n", tid);
printf("\n          n");
for(i=0; i<CAJEROS; i++)
{
    printf("\nCajero %i >> %x\n", i, idcajeros[i]);
}
nprocesos=CAJEROS;
while(nprocesos>0)
{
    id_buf=pvm_trecv(-1, 1, &tmout);
    if (id_buf>0)
    {
        pvm_bufinfo(id_buf, (int*)0, (int*)0, &id_rec); //
        pvm_upkint(&cod, 1, 1);
        pvm_upkstr(op);
        printf("\n          : %x\n", op, id_rec);

        rewind(pfich);
        fread(&per, sizeof(per), 1, pfich);
        while(!feof(pfich) && per.cod!=cod)
            fread(&per, sizeof(per), 1, pfich);
        if(per.cod==cod)
        {
            printf("\n          %i es %0.2f
euros\n", cod, per.saldo);
ok=0;
        }
        else
        {
            ok=1;
        }

        pvm_initsend(PvmDataDefault);
        pvm_pkint(&ok, 1, 1);
        pvm_send(id_rec, 1);
    }
    else
    {
        printf("\n          \n");
    }
    id_buf=pvm_trecv(-1, 2, &tmout);
    if(id_buf>0)
    {
        pvm_bufinfo(id_buf, (int*)0, (int*)0, &id_rec);
        pvm_upkint(&cod, 1, 1);
        pvm_upkfloat(&dinero, 1, 1);
        pvm_upkstr(op);
        printf("\n          %x\n", op, id_rec);

        rewind(pfich);
        fread(&per, sizeof(per), 1, pfich);
        while(!feof(pfich) && per.cod!=cod)
            fread(&per, sizeof(per), 1, pfich);
        if(per.cod==cod)
        {
            if(dinero<per.saldo)
            {
                printf("\n          %i %0.2f euros\n", cod, dinero);
                per.saldo-=dinero;
            }
        }
    }
}
}

```

Hay que matar los procesos creados
for(i=0; i<cc; ++i)
pvm_kill(idcajeros[i]);

```

        fseek(pfich, -(sizeof(per)), SEEK_CUR);
        fwrite(&per, sizeof(per), 1, pfich);
        printf("\n  \n");
        ok=0;
    }
    else
    {
        printf("\n .....%0.2f euros ....   %0.2f
euros.\n", dinero, per.saldo);
        ok=1;
    }
}
else
{
    ok=1;
}
pvm_initsend(PvmDataDefault);
pvm_pkint(&ok, 1, 1);
pvm_send(id_rec, 1);
}
else
{
    printf("  \n");
}

id_buf=pvm_trecv(-1, 4, &tmout);
if(id_buf>0)
{
    pvm_bufinfo(id_buf, (int*)0, (int*)0, &id_rec);
    pvm_upkint(&iter, 1, 1);
    pvm_upkstr(maquina);
    printf("\n\n      %x      %i operaciones\n", id_rec, iter);
    printf("\n      : %s\n\n", maquina);
    nprocesos--;
}
else
{
    printf("\n  \n");
}
}
printf("\n\n      \n");
i=1;
rewind(pfich);
fread(&per, sizeof(per), 1, pfich);
while(i<=10)
{
    printf("\n      %i      %0.2f\n", per.cod, per.saldo);
    fread(&per, sizeof(per), 1, pfich);
    i++;
}
fclose(pfich);
else
{
    printf("\n      ");
}
pvm_exit();
}

```

cajero.c

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include "pvm3.h"

#define OP 5
int GeneraAleatorio(int inf, int sup);
float GeneraCantidad(int inf, int sup);

```

```

main()
{
    int ptid,
        ok,
        operacion,
        cod,
        iter=0;
    float dinero;
    char buf[100];
    char maq[100];

    ptid = pvm_parent();

    while(iter<OP)
    {
        operacion=GeneraAleatorio(1,3);
        cod=GeneraAleatorio(1,10);
        pvm_initsend(PvmDataDefault);
        pvm_pkint(&cod,1,1);
        if(operacion==1)
            strcpy(buf,"C DE S ");
        if(operacion==2)
            strcpy(buf,"S D ");
        if(operacion==3)
            strcpy(buf,"I D ");
        if(operacion==2 || operacion==3)
        {
            dinero=GeneraCantidad(5,300);
            pvm_pkfloat(&dinero,1,1);
        }
        pvm_pkstr(buf);
        pvm_send(ptid,operacion);
        system("sleep 10");
        pvm_nrecv(ptid,1);
        pvm_upkint(&ok,1,1);
        if(ok==0)
        {
            iter++;
        }
    }

    gethostname(maq + strlen(maq),64);
    pvm_initsend(PvmDataDefault);
    pvm_pkint(&iter,1,1);
    pvm_pkstr(maq);
    pvm_send(ptid,4);
    pvm_exit();
}

int GeneraAleatorio(int inf, int sup)
{
    int n;
    srand((unsigned)time(NULL));
    n=inf+(int)((double)sup*rand()/(RAND_MAX+1.0));
    return n;
}

float GeneraCantidad(int inf, int sup)
{
    float n;
    srand((unsigned)time(NULL));
    n=inf+(float)((double)sup*rand()/(RAND_MAX+1.0));
    return n;
}

```