Primer Certamen: Tiempo: 15:40 hrs - 17:10 hrs.
Responder un problema por página

1.- (30 puntos) El administrador de un servidor de cuentas de usuarios desea evaluar el uso que ellos dan a su espacio de disco para decidir un aumento de espacio por cuenta (quota). A usted se le pide desarrollar quotaUsage, un script para listar el porcentaje de cuentas cuyos espacios ocupados de disco es superior al x%. x es un argumento del script.
Un ejemplo de ejecución es:
$ quotaUsage 90
15% de las cuentas ocupan más de 90% de su cuota.

Ayuda: Para consultar por uso de espacio en disco asignado, usted puede usar el comando quota. Éste arroja una salida del siguiente tipo:
$ quota -u agustin
Disk quotas for user agustin (uid 2502):

Filesystem	usage	quota	limit	grace	files	quota	limit	grace
/home	64052	102800	204800		101	0	0	

```
#! /bin/bash
n=0
overQuota=0
ifs_back=$IFS
IFS=":"
niscat passwd.org_dir |while read user rest
do
  echo $user
done > /tmp/t$$  # one user name per line
IFS=$ifs_back
while read user
do
  n=$(($n+1))
  echo "User = $user"
  quota -u $user | grep "/home" >> /tmp/q$$
  read usage quota rest < /tmp/q$$
  quotaTimesX=$(($quota*$1))   # to avoid decimals
  let usageTimes100=100*$usage
  if test $usageTimes100 -ge $quotaTimesX
  then
     overQuota=$(($overQuota+1))
  fi
done < /tmp/t$$
overQuota=$((100*$overQuota))
result=$(($overQuota/$n))
echo "$result% de las cuentas ocupan más de $1% de su cuota."
rm /tmp/t$$
rm /tmp/q$$
```

2.- (40 puntos) Cuando dos hebras acceden a un mismo dato y una de ellas lo modifica, otras podrían leer un valor inconsistente si no usamos mecanismos de exclusión mutua. Se desea medir el tiempo que toma en aparecer esa inconsistencia. Para ello se le pide hacer un programa que genere

dos hebras. En una usted intercambiará entre 1111111 y 2.222.222.000 el valor de una variable entera larga (long). La otra hebra leerá el valor de la variable, si éste es distinto de los dos valores indicados, enviará una alerta a pantalla y terminará la aplicación. Como este proceso podría tomar mucho tiempo, se le pide definir un tiempo máximo de ejecución en segundos ingresado como argumento en la línea de ejecución del programa. Transcurrido este tiempo el programa avisa que no encontró inconsistencias y termina.

b) Ponga mecanismos de exclusión para evitar la aparición de la inconsistencia. (indique el código y dónde lo insertaría, para no escribir todo su programa nuevamente)

```
#include <stdio.h>
#include <stdlib.h>       /* Puntaje: Programación Alarma (tiempo máximo ejecución): 8 pts. */
#include <signal.h>       /*            Programación thread que cambia el valor: 8 pts            */
#include <pthread.h>      /*            Programación thread que detecta inconsistencia: 8pts */
                          /*            Definición variable común como global: 6 pts */
#define value1 1111111LL          /* Parte a) 30 pts,   parte b) 10 pts.  */
#define value2 2222222000LL

int exclusion=0;
long long int value=value1;

static void sig_alrm(int signo) {
   printf("I can't find an inconsistancy\n");
   exit(0);
}

pthread_mutex_t mylock = PTHREAD_MUTEX_INITIALIZER;

void * changeValue(void * arg) {
   while (1) {
      if (exclusion)
        pthread_mutex_lock(&mylock);
      if (value==value1)
         value = value2;
      else
         value = value1;
      if (exclusion)
         pthread_mutex_unlock(&mylock);
   }
}

int main(int argc, char * argv[]) {
  int err;
  pthread_t tid;
  long long int readValue;
  int timeout;
  int* pStatus;

  timeout = atoi(argv[1]);

  if (signal(SIGALRM, sig_alrm) == SIG_ERR)
      exit(-1);
  err = pthread_create(&tid, NULL, changeValue, NULL);
  if (err != 0){
    printf("can't create thread \n");
```

```
   exit(0);
  }
  alarm(timeout);               /* start the timer */
  while (1){
     if (exclusion)
        pthread_mutex_lock(&mylock);
     readValue=value;
     if ((readValue!=value1) && (readValue!=value2)){
        printf("Inconsistency, the integer value is %lld\n",readValue);
        exit(0);
     }
     if (exclusion)
        pthread_mutex_unlock(&mylock);
  }
}
```

3.- (30 puntos) Haga un programa en C que muestre por pantalla el nombre real de un usuario a partir de su nombre de usuario en el sistema. Éste es pasado como argumento del programa. Su ejecución sería del tipo:
$ P3 agustin
Agustin J. Gonzalez

Ayuda: Considere el uso de finger, ejemplo:
$ finger agustin
Login: agustin Name: Agustin J. González
Directory: /home/agustin Shell: /bin/bash
….

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define BUFSIZE 100
int main(int argc, char * argv[]) {
   int i;
   FILE *pf;
   char line[BUFSIZE],command[BUFSIZE], * name;

   sprintf(command, "finger %s",argv[1]);
   if ((pf = popen(command, "r")) == NULL) {
      perror("popen");
      exit(1);
   }
   if (fgets(line, sizeof(line), pf) == NULL) {
      fprintf(stderr, "No output from finger command!\n");
      exit(1);
   }
   pclose(pf);
   strtok(line,":");
   strtok(NULL,":");
   name=strtok(NULL,":");
   printf("%s", name);
   exit(0);
}
```