Mariona Farré

Sistemes Operatius
Lab 3 . Entrenant als Replicants

Enigma 2
(LAB 3) SilverKey
El Capitol envia un agent per tal que interfereixi amb els elements subversius que intenten atacar el banc de temps del Capitol. Aquest agent compta amb un assassí professional anomenat Dr. Kill que té com arma letal una llista de senyals, estandarditzades per l'organització criminal POSIX, que permeten actuar sobre un TT.
Què senyal enviarà Dr. Kill per tal d'aturar temporalment un TT

Objectius d'aquesta sessió

- Creació i gestió de pipes
- Implementació d'un model client / servidor
- Concurrència i E/S: càlcul de valors òptims
- Gestió de signals. Comunicació entre processos.

Per tal d'entrenar als replicants en la seva tasca de robar temps del banc que hi ha al Capitol, crearem una simulació del Capitol que tindrà un comportament similar al que tindrà el Capitol real el dia de la **CompetiSIOP**.

La resistència ens ha proporcionat un «generador de Banc de Temps» anomenat **gendummy [S/M/L]**, que ens permet crear un banc de temps fictici per tal d'entrenar als replicants. Una vegada activat el generador, es crea un Banc de Temps anomenant **dummy.dat** de mida:
•100MB de temps si es crida amb argument S (small)
•500MB de temps si es crida amb argument M (medium)
•1GB de temps si es crida amb argument L (large)

A mes, també ens proporcionen un altre dummy, el dummy3 que substituirà a l'anterior. Aquest conté un conjunt de funcions que ens permetran fer les següents accions:
dummy_checkfile(fd): comprova que sigui un banc de temps correcte.
dummy_calc (buffer, bytesllegits): cada vegada que un replicant roba un bloc de N unitats de temps, l'ha de processar en aquesta funció per a convertir-ho ens temps de vida real.
dummy_testing (acum, ENIGMA2,TEAMNAME,BRONCEKEY): acumulat dels bancs de temps, solucion enigma,nom, ens diu si, una vegada finalitzat l'atac i processat tot el temps obtingut, el processament s'ha realitzat correctament.
dummy_init(buffer, N): prepara el contenidor de memòria del replicant
dummy_end(): permet que el replicant surti del Capitol sense deixar cap empremta. A més envia el seu codi de finalització al TT per a que el pugui recuperar.
•**Ready()**: permet saber al TT que el Capitol està preparat per enviar informació.
Comunicacio entre TT i Capitol

1. Capitol

Per tal de simular els protocols de comunicació entre client/servidor, el Capitol ha de comunicar al TT que es troba preparat per enviar informació.
Una vegada feta la comunicació, executa el següent codi:

while (read(0, &c, 1) > 0) write(1, &c, 1);
exit(0);

mentre hi hagui el input (read(0&c,1)) anar escrivint per pantalla
el capitol es un programa que es convertiex en un proces
El capitol i TT han s'estar comunicats- shan de comunicar els dos processos
-S'han de comunicar amb SIGNAL
(funcio ready comprova la relacio) (capitol fitxer fill del TT)
una vegada preparat el capitol s'executara el codi del capitol

a) Identifica l'enunciat amb els continguts de l'assignatura
b) Realitza un pseudocodi que expliqui clarament, en termes de l'assignatura, com es prepara el Capitol.
c) Crea el Capitol.

2. Time Thief
El TT es l'encarregat de crear el Capitol simulat i un canal de comunicació entre ells dos. El TT obrirà el banc de temps i permetrà que el Capitol el deixi accessible al canal de comunicació. Això simularà el comportament real en el que un agent autoritzant demana temps al banc de temps que hi ha al Capitol.
A continuació el TT ha de crear els replicants i esperar que acabin d'atacar el banc de temps que es troba al canal de comunicació.
Evidentment, els replicants han de tenir accés a aquest canal.

es vol passar informació - replicants roben la informació
capitol-banc de temps-replicant entra i roba el temps que vol
Els replicants necessita un canal de comunicacio per passar el temps-funcio PIPE

Tal i com ja hem fet, a mida que els replicans vagin finalitzant la seva missió, el TT ha de recuperar el codi de finalització de cadascú (valor entre 0 i 127) i l'acumularà amb la resta de codis dels altres replicants. Aquest valor serà entregat a la funció **dummy_testing(acum, ENIGMA2,TEAMNAME,BRONCEKEY)** per saber si el processament s'ha realitzat correctament. A més, serà necessari entregar a aquesta funció , el nom del vostre equip i la clau obtinguda en el lab anterior.
En cas que qualsevol dels replicants no hagi pogut completar la seva missió amb èxit, el TT avortarà la missió d'atac i enviarà un missatge informatiu.
En tot moment hem d'aprofitar el dummy3, proporcionat per la resistència, que ens facilitarà diferents tasques a desenvolupar.
El codi del TT ha de contenir els següents valors:
#define TEAMNAME "" // insert between "" your team name
#define ENIGMA2 "" // insert between "" solution of enigna 2
#define BRONCEKEY "" // insert BRONCEKEY between ""
d) Identifica l'enunciat amb els continguts de l'assignatura
e) Realitza un pseudocodi que expliqui clarament, en termes de l'assignatura, com es prepara el TT per l'atac al banc de temps.

f) Modifica el TT amb les noves accions.

3. Replicants

Reviseu que el replicant funciona correctament amb els canvis que ha fet el TT (ara no ataca directament a un banc de temps sinó que ho far a traves d'un canal de comunicació) Adapteu el codi a les noves funcions dummy3.

Comproveu el correcte funcionament de l'aplicació usant un banc de temps dummy.dat de mida SMALL o MEDIUM (per anar depurant el codi). Comproveu que funciona correctament a partir dels missatges generats per les rutines dummy3.

Un cop comprovat el correcte funcionament, busqueu els valors òptims de nombre de Replicants / mida del bloc de temps a robar per a que el temps d'execució i processament de dades sigui el menor. Podeu saber el temps d'execució d'un programa, usant la comanda:

time ./programa

a) Heu de lliurar: Una taula en la que es vegi quins són aquests valors òptims.

NOTA: es possible que, si s'atura l'execució de forma forçada (Ctrl+C, per exemple) dels processos d'aquest lab, es perdi el cursor del terminal. Per restaurar-ho podeu fer:

```
# setterm -cursor o
```

COMANDES:

```
gedit capitol.c
gcc capitol.c -o capitol
gcc TT.c -o TT -L. -ldummy3
gcc replicant.c -o replicant  -L. -ldummy3
```