

Nume și specializare:

Sisteme de Operare Avansate

21 ianuarie 2013

Timp de lucru: 60 de minute

Notă: Toate răspunsurile trebuie justificate

1. Câte întreruperi poate trata un sistem de operare la un moment dat?
2. Ce problemă concretă rezolvă un futex față de primitivele de sincronizare clasice? Ce problemă nouă introduce?
3. Dați un exemplu de situație în care RCU se comportă mai prost decât sincronizarea clasică?
4. Pentru un atac cu page-fault-uri, scrieți o bucată de cod care să poată exploata o vulnerabilitate a unei funcții cu un singur argument.
5. Descrieți virtualizarea apelului de sistem `getpid()` invocat de un proces care rulează în UML.
6. Explicați de ce este para-virtualizarea preferată în Xen?
7. Precizați un avantaj și un dezavantaj al folosirii analizei statice bazată pe Java reflection în fața analizei dinamice pe cod Android pentru realizarea unei tabele de mapare de permisiuni (*permission map*).
8. De ce nu poate fi folosit modelul de sincronizare al RFS în sisteme de versionare de uz generic (git, SVN, etc.)?
9. Care este avantajul unui checkpoint bazat pe fork, față de unul clasic?
10. Fie un driver GPU implementat într-un sistem L4Android cu interfață HAL. Ce componentă a acestuia se va găsi în nucleul Android al mașinii guest și ce componentă în HAL?
11. De ce este atacul următor un atac *non-control data* și ce obține atacatorul folosindu-l:

```
void serve (int fd)
{
    char *name = globMyHost;
    char cl name[64];
    char svr reply[1024];

    read(fd,cl name,128);
    sprintf(svr reply, "hello %s, I am %s", cl_name, name);
    svr_send(fd,svr reply,1024);
}
```

În conformitate cu ghidul de etică al Catedrei de Calculatoare, declar că nu am copiat și nu voi copia la această lucrare. De asemenea, nu am ajutat și nu voi ajuta pe nimeni să copieze la această lucrare.

Nume și grupă:

Semnătură:.....