

# Project 4: Kernel Hacking

Tekijät: Sara Heikkinen (567012) ja Alex Karonen (566770)

## Lyhyt yhteenveto tehtävästä:

Käyttöjärjestelmään xv6 luotiin systeemikäsky getCount(), jonka avulla pystytään laskemaan kuinka monta kertaa terminaalin aukiollessa systeemikutsua sys\_read on kutsuttu. Käskeyn luotiin mahdollisuus nollata laskuri terminaalin aukiollessa ja tällöin ohjelma aloittaa laskemisen lähtien nolasta.

## Ohjelmaa voidaan ajaa kahdella eri tavalla:

getCount (lasketaan kuinka monta kertaa systeemikutsua sys\_read on kutsuttu terminaalin aukiollessa)

getCount -r (parametri -r nollaa laskurin ja ilmoittaa käyttäjälle että laskuri on nollattu)

Harjoitustyössä luotiin uusi systeemikutsu getCount(), joka laskee kuinka monta kertaa systeemikutsua sys\_read on kutsuttu. Apua systeemikutsun luontiin katsottiin videosta <https://www.youtube.com/watch?v=21SVYiKhcwM>. Systeemikutsu piti määrittellä todella moneen tiedostoon, joten videota seuraamalla varmistettiin, että määrittelimme omamme jokaiseen tarvittavaan tiedostoon.

Inspiraatiota systeemikutsun laskemiseen saatiin videosta

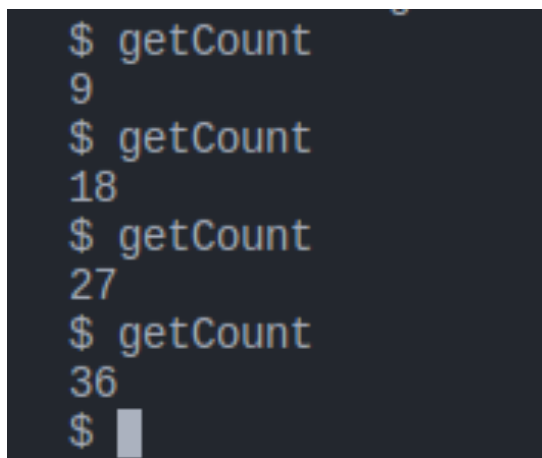
[https://www.youtube.com/watch?v=Hl\\_t6BEBJko](https://www.youtube.com/watch?v=Hl_t6BEBJko), sekä dokumentaatiosta

<https://gist.github.com/bridgesign/e932115f1d58c7e763e6e443500c6561>. Esimerkkejä seuraamalla ymmärsimme miten eri tiedostot kytkeytyvät toisiinsa, sekä sen, mikä tehtävä milläkin määrittelyllä on.

Laskurin nollaus implementoitiin projektiin itsenäisesti.

Jos kutsutaan vain getCount() funktiota, saadaan joka kerta yhdeksän sys\_read komennon kutsua lisää. On loogista, että käyttöjärjestelmä ajaa saman systeemikutsun joka kerta samalla tavalla ja tällöin sys\_read komentoa kutsutaan aina vakiomäärä.

**Esimerkki 1:** Kutsutaan getCount() kutsua muutama kerta ja huomataan, että jokaisella kerralla laskuri kasvaa vakiomäärän



```
$ getCount
9
$ getCount
18
$ getCount
27
$ getCount
36
$
```

**Esimerkki 2:** kutsutaan jotakin toista kutsua ja laskuri ei kasva enää saman yhdeksän verran

```
$ getCount
36
$ ls
.          1 1 512
..         1 1 512
README    2 2 2286
cat        2 3 16336
echo       2 4 15192
forktest   2 5 9496
grep       2 6 18556
init       2 7 15776
kill       2 8 15220
ln         2 9 15072
ls         2 10 17704
mkdir     2 11 15320
rm         2 12 15296
sh         2 13 27940
stressfs   2 14 16208
usertests  2 15 67316
wc         2 16 17072
zombie     2 17 14888
getCount   2 18 15064
console    3 19 0
$ getCount
81
$
```

**Esimerkki 3:** Kutsutaan getCount() kutsua parametrilla -r ja saadaan ilmoitus, että laskuri on nollattu. Seuraavan kerran kun kutsutaan getCount() kutsua, saadaan sys\_read komentojen kutsumääräksi taas yhdeksän

```
$ getCount
81
$ getCount -r
Counter reseted! Counter is 0
$ getCount
9
$
```