

Styresystemer og multiprogrammering (OSM)

- G1

Dennis Bøgelund Olesen - 060593 - cwb759

Emil Lagoni - 051290 - frs575

Erik David Allin - 171292 - smt504

17. Februar 2013

Task 1

Task 2

Filer involveret: fs/**io.c**, fs/**io.h** samt tests/**readwrite.c**.

I denne opgave, udnyttede vi os af typen `device` fra `drivers`, som tillod os at bruge kernel-kaldene `read` og `write`. Vi skulle altså lave en driver pointer.

Da `device`-strukturen har et generisk `device` i sin struktur, kan vi udnytte den GCD vi har lavet. Vi kan nemlig se af GCD, at den har henholdsvis `read` og `write`, som gør nøjagtigt det vi ønsker.

I forbindelse med dette bruger vi kernel assert til at sikre os, at vi peger på et `device`.

At **io.c** og **io.h** ligger i mappen `fs`, er taget fra `buenos roadmap`, som har inddelt `read` og `write` som systemkald, der relaterer til filsystemer. som beskrevet i `buenos roadmap`, side 44.

a Testing

For at teste `readwrite`, lavede vi filen `readwrite.c` i mappen `tests0`.

Efter at have compilet denne, lavede vi:

```
util/tfstool create fyams.harddisk 2048 disk1
```

og

```
util/tfstool write fyams.harddisk tests/readwrite readwrite
```

Når dette er lavet. kan testen køres med kommandoen:

```
fyams-sim buenos 'initprog=[disk1]readwrite'
```

Når dette er startet er det muligt at taste i terminalen, hvorefter `read` så læser det du skriver, og `write` skriver det ud til terminalen igen.

I vores test er der brugt en int buffer. Dette betyder, at alt fylder 4 bytes, så der kan altså ikke læses 63 chars, men derimod kun en fjerdedel.

C kan dog sagtens se chars som integers.