

# Guió TFC

Esteve Espuña Sargatal *St11227*

10 de gener de 2005

# 1 Disseny i implementació d'un kernel monolític

L'enfoc del treball serà anar construint un kernel monolític de forma incremental, pas a pas, abans de cada pas una petita introducció teòrica, background, disseny escollit i implementació. El guió a seguir serà el següent:

- Introducció
  - Explicar de forma general de què tractarà el treball subratllar el caràcter didàctic del mateix.
  - Objectiu final.

- Disseny General

S'explica què és un kernel i l'enfoc més ampli del disseny, es dona una perspectiva dels kernels actuals. A partir d'aquí s'explica perquè es realitza un kernel monolític sobre intel X86, i s'explica l'enfoc de portabilitat.

- Què és un kernel
  - Classificació dels kernel
  - Principals Kernel Existents, ús i quota de mercat de cada un
  - Arquitectura del kernel a implementar
  - Arquitectura Destí
  - Portabilitat
- Entorn de Treball inicial

Abans de cada capítol ( si és necessari ) explicaré el conjunt d'eines necessàries per poder seguir el capítol sense problemes, l'entorn de treball s'anirà construint a mesura que s'avanci en la construcció del kernel. En aquest primer es necessita només un compilador d'assembler i l'emulador de x86.

- Introducció: Indicar que totes les eines que s'utilitzaran seran GPL i que pot ser utilitzat amb qualsevol plataforma ( Win, Linux, MacOSX ).

- Compilador Nasm
  - Grub Disk
  - Emulador X86 Bochs
- Estructura de les fonts

Explicaré el layout de l'arbre de fitxers del projecte, a mesura que es vagi ampliant el kernel aniré ampliant l'arbre de directoris. Aquí ha d'incloure, arrel, directori pel boot, directori kernel, directori includes.

- Procés de Boot

S'ha de recordar que tractarem de X86. Hem d'indicar totes les possibilitats que tenim des que es pulsa el botó fins que es carrega el nostre kernel en memòria.

- Introducció
  - Què passa a partir de pulsar el botó d'engegar l'ordinador ( POST )
  - Mode real, protegit, unreal i VM86
  - Què és un BootLoader
  - Estàndard Multiboot
  - Perspectives de disseny: Dos enfoc, fer el bootloader, utilitzar un bootlader MultiBoot
  - Disseny i implementació d'un mínim bootloader
  - Grub Bootloader : Avantatges i mínim exemple
- Entorn de Treball : Compilador Gcc
    - Introducció: Què és el Gcc, perquè el necessitem i les features que utilizem
    - Opcions de compilació
    - Opcions de linkatge fitxer de configuració de l'ld

- Cross Compiler ( Què passa si no tenim un X86 o la configuració del nostre no és la correcte )

- Kernel Inicial

Realitzem dos kernels petits que treuran per pantalla Hello World, un serà un binari pla i l'altre un ELF, finalment decidim usar ELF.

- Introducció
- Binari pla
- ELF
- Decisió de disseny: ELF
- Abandonem el nostre bootloader per utilitzar grub, realitzem tot el mode de vídeo.

- Memòria, Segmentació Paginada ( Explicació amb el bootloader que em realitzat )

- Introducció: Diferents modes de memòria
- Elecció de segmentació paginada
- Explicació Segmentació ( 3 segments solapats CS, DS, SS )
- Explicació 2 nivells pagines
- Mapa de memòria en el moment de carregar el kernel. ( Diagrama com queda Data Codi ... )
- Implementem el disseny en el nostre kernel ELF utilitzant Grub

- Interrupcions

Explico molt per sobre què son les interrupcions i com les tractem en les primeres etapes, és necessari per poder explicar el sistema de drivers.

- Sistema de Drivers

- Disseny d'una API de drivers

- Estructura de directori pel sistema de drivers
  - Drivers de sistema i video
  - Implementació de tot el que s'ha explicat
- Multitasca