



LAB 5

SUPPORTO PER FILE SYSTEM

PROGRAMMAZIONE DI SISTEMA (JZ-ZZ)

2022/23

ANDREA PORTALURI

OBIETTIVI DEL LABORATORIO

- Implementazione della system calls open/close
- Completamento system calls read/write
- (opzionale) Argomenti al main

ATTENZIONE: per questo laboratorio è richiesto il completamento dei laboratori 2 e 4, aver compreso il flow di implementazione di una system call e la gestione e sincronizzazione di un processo.

FILE SYSTEM IN OS161

In OS161 manca il supporto per il file system, inteso come insieme delle system calls che forniscono operazioni sui files. Tale supporto, oltre alla realizzazione delle singole funzioni, necessita opportune strutture dati che permettano la ricerca dei files e la loro identificazione.

Si richiede la realizzazione in forma parziale di `open`, `close`, `read` and `write` dopo aver familiarizzato con le strutture dati e le funzioni kernel coinvolte.

FILE SYSTEM IN OS161

- **Virtual File System (VFS)** in `kern/vfs.c` e `kern/include/vfs.h`, dove la `struct vnode` rappresenta un file e le funzioni `vfs_open` e `vfs_close` servono ad aprirlo e chiuderlo (si veda `runprogram`).
- **Modulo `uio`** per il trasferimento tra file, memoria kernel e user. Le funzioni di principale interesse sono `uio_kinit` e `uiomove`

Si consiglia di osservare la gestione IO in `runprogram/load_elf/load_segment` delle funzioni coinvolte.

Per il test di `open/close` si consiglia di utilizzare `testbin/filetest`

OPEN/CLOSE SYSTEM CALLS

Si consiglia di realizzare `open` e `close` negli stessi file già usati per `read` e `write`. Per un dato processo user, occorre generare una tabella di puntatori a `vnode`. La costante `OPEN_MAX` (`limits.h`) definisce il massimo numero di file aperti per un processo. Ad ogni file aperto fa assegnato il file descriptor (intero non negativo).

Attenzione: `stdin`, `stdout` e `stderr`, a meno che siano rediretti su file, sono associati alla console (`kern/dev/generic/console.c`) gestita mediante `kprintf` (che indirettamente chiama `putch`) e `kgets` (che chiama `getch`). La tabella dei file va gestita in modo tale che la presenza o meno di un file su cui sia stato rediretto `stdin`, `stdout` o `stderr`, determini il tipo di IO da effettuare (a console o a file).

PASSAGGIO ARGOMENTI AL MAIN

Per passare gli argomenti (`argv`) ad un programma utente occorre caricare gli argomenti nello spazio di indirizzamento dello stesso. Per comprendere come funziona il passaggio di argomenti, si consideri `testbin/tail` che vuole due argomenti.

Attenzione: un `char` può essere collocato in un qualunque indirizzo virtuale, mentre un puntatore deve essere a un indirizzo multiplo di 4 (padding).