

# Sistemi Operativi - Tutoraggi

Laurea in Ingegneria Informatica

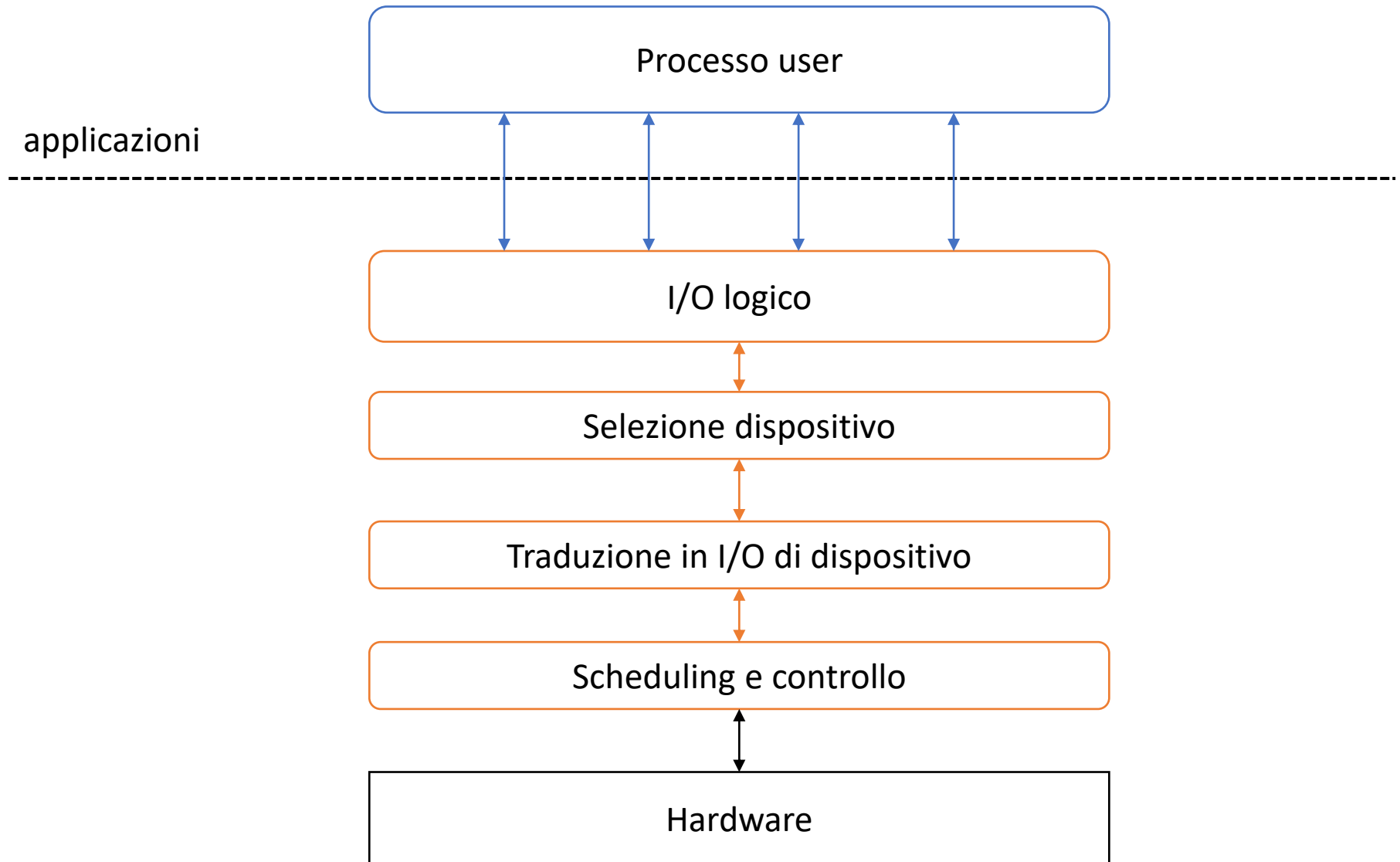
Università Tor Vergata

Tutor: Romolo Marotta

Docente del corso: Francesco Quaglia

## Virtual File System

# Modello di organizzazione moduli di I/O



# Operazioni su file

## Creazione

- Allocazione del record di sistema e inizializzazione del file

## Eliminazione

- Deallocazione del record di sistema e rilascio di tutti i record allocati per il file di interesse

## Apertura

- Inizializzazione di una sessione e di un relativo indice per accessi al file nella sessione

## Chiusura

- Distruzione della sessione, dei suoi metadati e dell'indice

## Scrittura/Lettura

- Accesso a partire dal record a cui l'indice fa riferimento

## Riposizionamento (seek)

- Aggiornamento dell'indice

## Troncamento

- Distruzione di record del file

# Esercizio 1

- Nei sistemi operativi UNIX, `/dev/urandom` è un dispositivo a caratteri virtuale in grado di generare numeri casuali. Nello specifico, l'operazione di lettura dal relativo file produce byte casuali.
- Scrivere un programma C che:
  - prende come parametri da linea di comando: un numero `N` e una stringa `S` da usare come nome del file da creare;
  - crea un file `S` contenente `N` byte randomici;
  - utilizza il dispositivo `/dev/random` come sorgente di numeri pseudo-casuali.

## Esercizio 2

Creare un file di  $2^{15}$  interi di tipo short utilizzando la soluzione dell'esercizio 1

Scrivere un programma che dato un file di  $2^{15}$  interi di tipo short:

- Crea tanti thread quanti sono i core disponibili
- Calcola minimo e massimo short contenuto nel file

# Esercizio 2

- Nel fornire una soluzione rispettare i seguenti vincoli:
  - ciascun thread è pinnato su uno specifico core
  - ciascun thread legge una o più porzioni del file una ed una sola volta
  - ciascun intero non può essere letto da più di un thread
  - ciascun thread può allocare memoria per al più 512 byte
  - è ammesso allocare di variabili globali (data) e locali (stack) per memorizzare tipi primitivi (puntatori, int, short, char, long, etc.) per al più 128 byte
  - non è ammesso l'utilizzo di costanti per indicare la taglia del file

# Esercizio 3

- In aggiunta, i thread collaborano ad invertire la posizione di ciascun short all'interno del file
- Primo short in ultima posizione, ultimo short in prima posizione, secondo short in penultima posizione, penultimo short in seconda posizione, e così via.
- Senza utilizzare un file aggiuntivo e senza utilizzare un buffer per accomodare l'intero file in memoria.

# Esercizio 4

- In aggiunta al minimo e massimo, identificare gli interi distinti contenuti nel file e
- Per risolvere il problema assumere che:
  - il file non entra interamente in RAM (e.g. 60GB)
  - i thread possono allocare fino a 8KB di memoria