ESD Electronic Systems Design

Synthesis Verification Testing Power Communication

Componenti di un sistema operativo

Dipartimento di Informatica

Università di Verona, Italy

Componenti di un S.O.

- Gestione dei processi
- Gestione della memoria primaria
- Gestione della memoria secondaria
- Gestione dell'I/O
- Gestione dei file
- Protezione
- Rete
- Interprete dei comandi

Gestione dei processi

- Processo = programma in esecuzione
  - Necessità di risorse
  - Eseguito in modo sequenziale un'istruzione alla volta
  - Processi del S.O. vs. processi utente
- Il S.O. è responsabile della
  - Creazione e distruzione di processi
  - Sospensione e riesumazione di processi
  - Fornitura di meccanismi per la sincronizzazione e la comunicazione tra processi

Gestione della memoria primaria

- Memoria primaria conserva dati condivisi dalla CPU e dai dispositivi di I/O
  - Un programma deve essere caricato in memoria per poter essere eseguito
- Il S.O. è responsabile della
  - Gestione dello spazio di memoria (quali parti e da chi sono usate)
  - Decisione su quale processo caricare in memoria quando esiste spazio disponibile
  - Allocazione e rilascio dello spazio di memoria

Gestione della memoria secondaria

- Memoria primaria è volatile e “piccola”
  - Indispensabile memoria secondaria per mantenere grandi quantità di dati in modo permanente
- Tipicamente uno o più dischi (magnetici)
- Il S.O. è responsabile della
  - Gestione dello spazio libero su disco
  - Allocazione dello spazio su disco
  - Scheduling degli accessi su disco

Gestione dell'I/O

- Il S.O. nasconde all'utente le specifiche caratteristiche dei dispositivi di I/O
- Il sistema di I/O consiste di
  - Un sistema per accumulare gli accessi ai dispositivi (buffering)
  - Una generica interfaccia verso i device driver
  - Device driver specifici per alcuni dispositivi

Gestione dei file

- Le informazioni sono memorizzate su supporti fisici diversi (dischi, DVD, memory-stick, …) controllati da driver con caratteristiche diverse
- File = astrazione logica per rendere conveniente l'uso della memoria non volatile
  - Raccolta di informazioni correlate (dati o programmi)
- Il S.O. è responsabile della
  - Creazione e cancellazione di file e directory
  - Supporto di primitive per la gestione di file e directory (copia, sposta, modifica, …)
  - Corrispondenza tra file e spazio fisico su disco
  - Salvataggio delle informazioni a scopo di backup

Protezione

- Meccanismo per controllare l'accesso alle risorse da parte di utenti e processi
- Il S.O. è responsabile della:
  - Definizione di accessi autorizzati e non
  - Definizione dei controlli da imporre
  - Fornitura di strumenti per verificare le politica di accesso

Rete (Sistemi distribuiti)

- Sistema distribuito = collezione di elementi di calcolo che non condividono né la memoria né un clock
  - Risorse di calcolo connesse tramite una rete
- Il S.O. è responsabile della gestione “in rete” delle varie componenti
  - Processi distribuiti
  - Memoria distribuita
  - File system distribuito
  - …

Interprete dei comandi (Shell)

- La maggior parte dei comandi vengono forniti al S.O. tramite “istruzioni di controllo” che permettono di:
  - Creare e gestire processi
  - Gestire l’I/O
  - Gestire il disco, la memoria, il file system
  - Gestire le protezioni
  - Gestire la rete
- Il programma che legge ed interpreta questi comandi è l’interprete dei comandi
  - Funzione: leggere ed eseguire la successiva istruzione di controllo (comando)

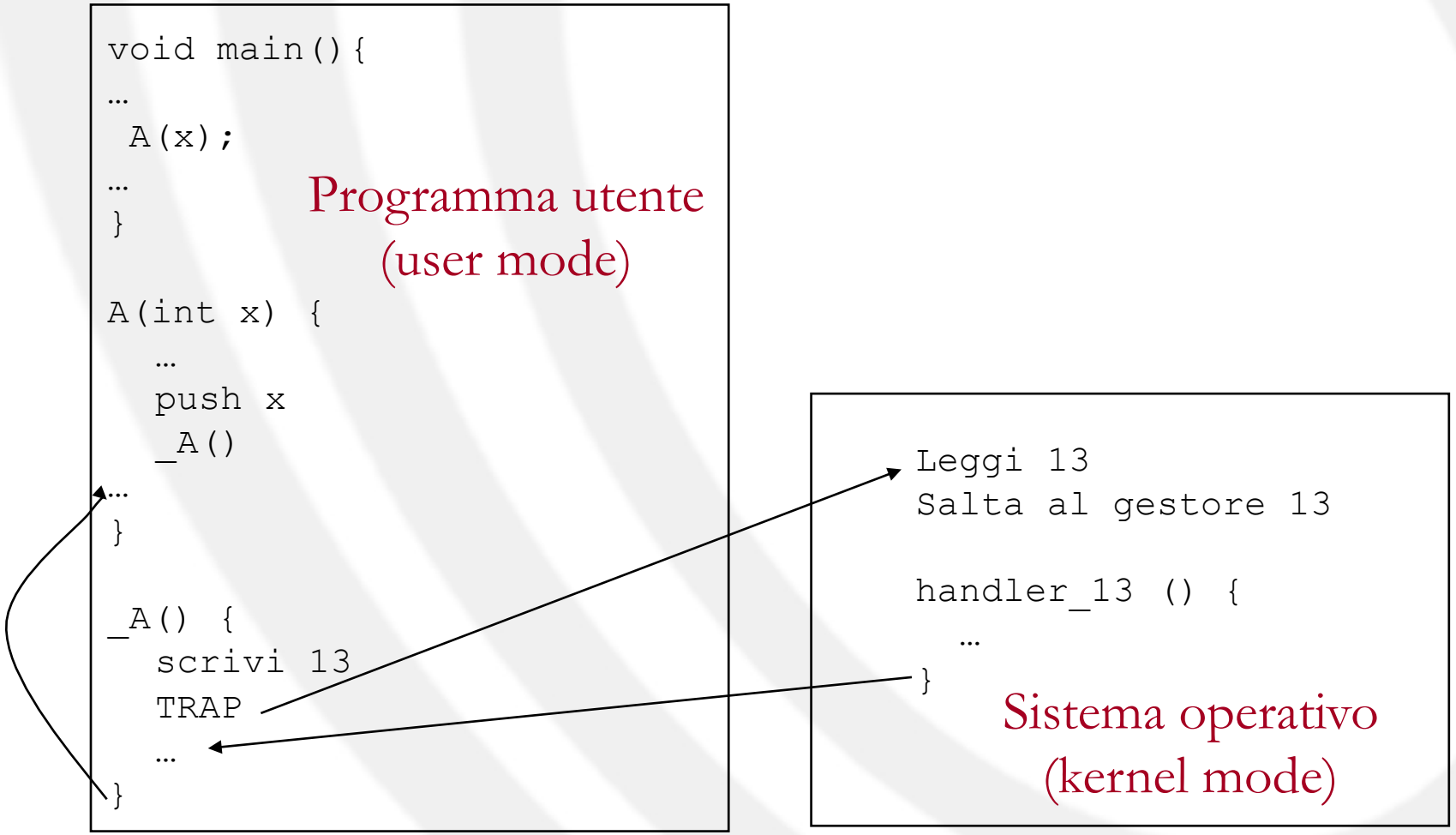
System Call

- L'utente usa la shell, ma i processi?
  - Le system call forniscono l'interfaccia tra i processi e il S.O.
- Opzioni per la comunicazione tra il S.O. e un processo:
  - Passare i parametri (della system call) tramite registri
  - Passare i parametri tramite lo stack del programma
  - Memorizzare i parametri in una tabella in memoria
    - L'indirizzo della tabella è passato in un registro o nello stack

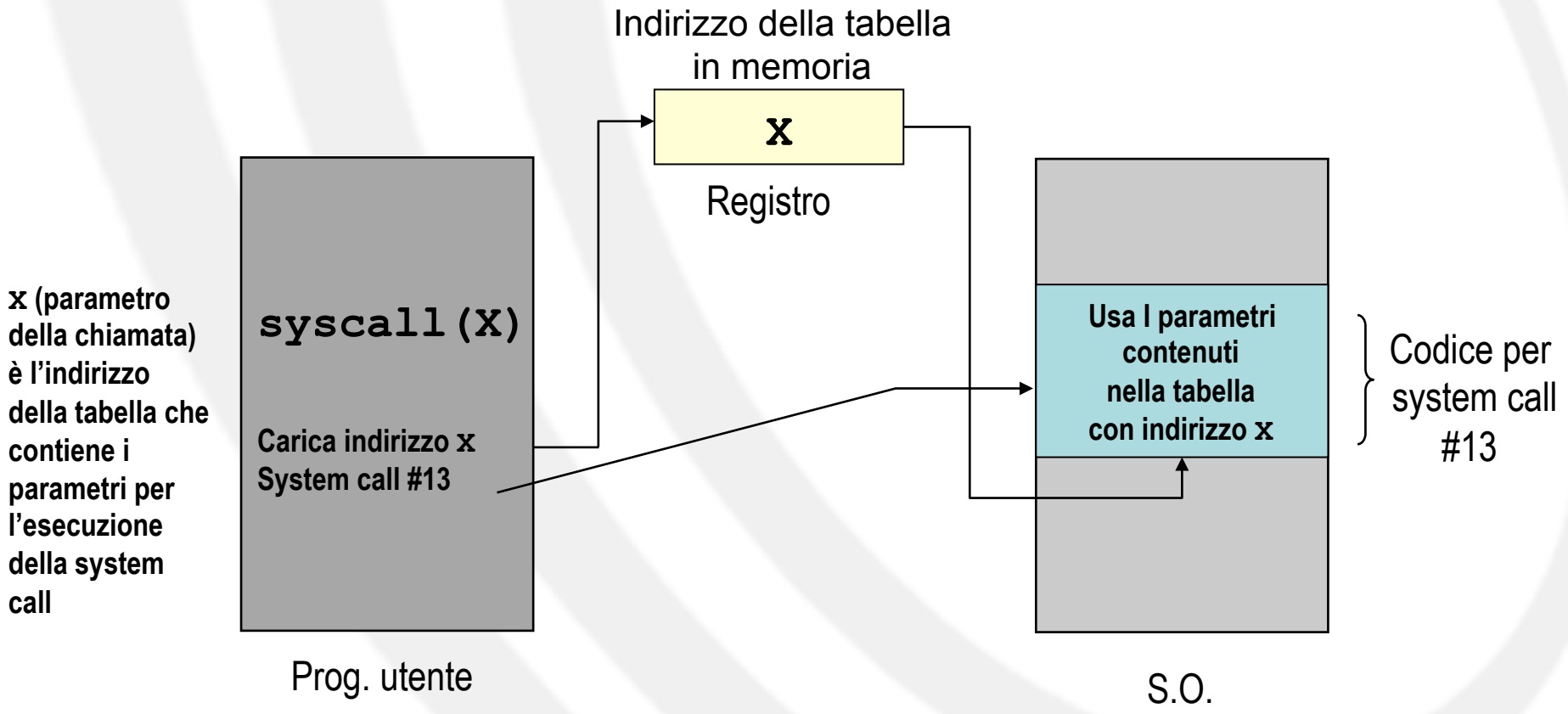
Passaggio di parametri nello stack

- Chiamata alla funzione di libreria A(x)
  - Parametro x nello stack
  - Invocazione della vera system call _A corrispondente ad A
  - _A mette il numero di system call in un punto noto al S.O.
  - _A esegue una TRAP (interruzione non mascherabile)
    - Effetto: passaggio da Modo User a Modo Kernel
  - Inizia l'esecuzione ad un indirizzo fisso (gestore interrupt)
  - Il S.O., in base al numero di system call, smista la chiamata al corretto gestore che viene eseguito
  - Una volta terminato, il controllo viene restituito al programma di partenza (funzione di libreria A())

Passaggio di parametri nello stack



Passaggio di parametri tramite tabella



Programmi di sistema

- La vista utente delle operazioni di un sistema avviene tipicamente in termini di programmi di sistema (e non di system call)
  - Gestione/manipolazione dei file (crea, copia, cancella, …)
  - Informazioni sullo stato del sistema (data, memoria libera, …)
  - Strumenti di supporto alla programmazione (compilatori, assemblatori, …)
  - Formattazione documenti
  - Mail
  - Programmi di gestione della rete (login remoto, …)
  - Interprete dei comandi
  - Utility varie

Riassumendo…

- Servizi di un S.O.:
  - Esecuzione di programmi
  - Operazioni di I/O
  - Manipolazione del file system
  - Comunicazione
    - Memoria condivisa
    - Scambio di messaggi
  - Rilevamento di errori (logici e fisici)
  - Allocazione delle risorse
  - Contabilizzazione delle risorse
  - Protezione e sicurezza