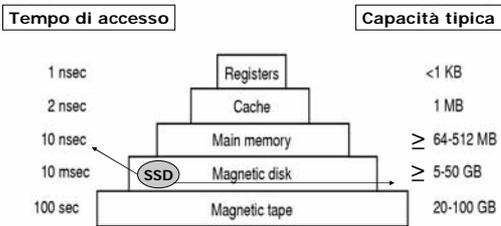


## Gerarchia fisica di memoria – 1



Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 95/121

## Gerarchia fisica di memoria – 2

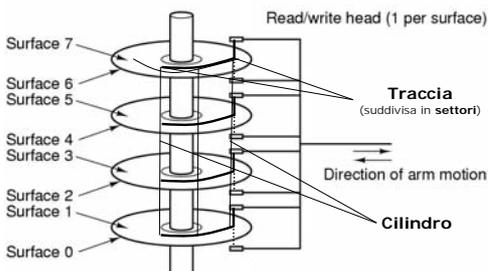
- La **cache** è suddivisa in blocchi chiamati *line* con ampiezza tipica 64 B
  - Costi e benefici: *miss, hit*
  - Coerenza della memoria: *write-through, copy-back*
- I **dischi magnetici** hanno capienza 100 volte superiore e costo/*bit* 100 volte inferiore rispetto alla RAM
  - Tempo di accesso 1.000 volte peggiore
- La tecnologia SSD (**solid state drive**) offre una alternativa interessante
  - Capienza paragonabile ai dischi
  - Tempo d'accesso nell'ordine della RAM
    - Grazie all'assenza di parti meccaniche
  - Costo/*bit* > 30 volte superiore ai dischi

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 96/121

## Gerarchia fisica di memoria – 3

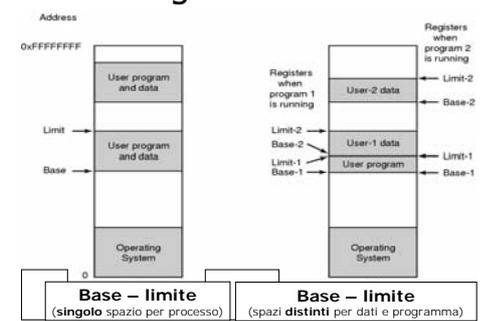


Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 97/121

## Vista logica della RAM – 1



Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 98/121

## Vista logica della RAM – 2

- Nella sua forma più rudimentale la ripartizione della RAM tra processi distinti utilizza 2 registri speciali
  - Base** e **limite**, i cui valori formano parte importante del contesto del processo
  - L'allocazione di un processo in RAM richiede **rilocazione** della sua **memoria virtuale**
    - Amplia  $2^N$  Byte dove N è il numero di *bit* usati per un indirizzo
- In generale la gestione dello spazio di memoria virtuale dei processi utilizza un dispositivo **MMU** (*Memory Management Unit*) logicamente interposto tra CPU e memoria
  - Sotto la responsabilità del S/O

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 99/121

## Vista logica della RAM – 3

- Il S/O assegna una porzione di RAM a ogni processo
- Un processo emette indirizzi logici **I** all'interno della propria memoria virtuale
  - Indirizzi contigui e relativi a una base logica "zero"
- L'indirizzo logico **I** va tradotto in un indirizzo fisico **M**
  - Traduzione (**rilocazione**) effettuata dalla MMU come  $M = \text{Base} + I$
  - Se  $M < \text{Limite}$  allora l'indirizzo si trova davvero nella zona di RAM assegnata al processo
    - Altrimenti va cercato in memoria secondaria
    - La MMU deve allora effettuare un altro calcolo su **M** per capire la zona di disco da dove prelevarlo
    - Il caricamento di dati in RAM può comportare il rimpiazzo di altri dati
      - La RAM ha capacità limitata mentre i processi sono tanti e "esososi"

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 100/121

## Macchina virtuale – 1

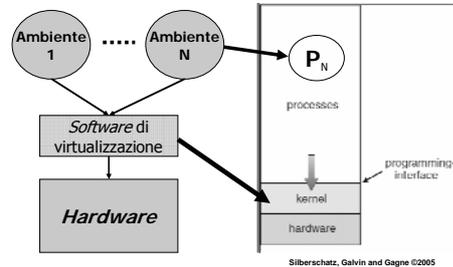
- Alla base del concetto stesso di multiprogrammazione
  - Condividere trasparentemente tra molti utenti una risorsa progettata per uso singolo
    - L'intero elaboratore o una parte di esso
    - Compito originario del sistema operativo
- La virtualizzazione porta grandi benefici
  - Fino al punto di consentire l'esecuzione simultanea di diversi sistemi operativi con le loro applicazioni
    - Richiede uno strato *software* dedicato tra l'*hardware* e i sistemi operativi ospitati oppure sopra il sistema operativo scelto come ospite

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 101/121

## Macchina virtuale – 2

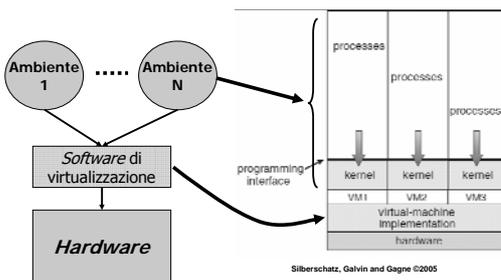


Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 102/121

## Macchina virtuale – 3

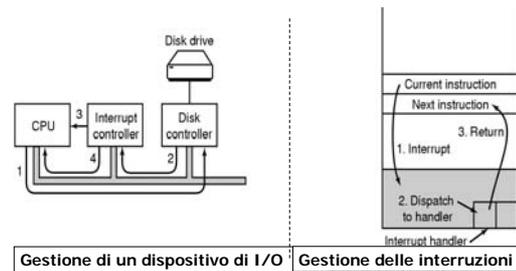


Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 103/121

## Gestione dell'I/O – 1



Gestione di un dispositivo di I/O

Gestione delle interruzioni

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 104/121

## Gestione dell'I/O – 2

- L'uso delle interruzioni per l'interazione con i dispositivi di I/O risparmia il ricorso al *polling*
- L'interazione tipica avviene in 4 passi successivi
  1. Il gestore del dispositivo programma il controllore di dispositivo scrivendo nei suoi registri di interfaccia
  2. Il controllore agisce sul dispositivo e poi informa il controllore delle interruzioni
  3. Il controllore delle interruzioni asserisce un valore (*pin*) di notifica verso la CPU
  4. Quando la CPU si dispone a ricevere la notifica il controllore delle interruzioni comunica anche l'identità del dispositivo
    - Così che il trattamento dell'interruzione sia attribuito al gestore appropriato

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 105/121

## Gestione dell'I/O – 3

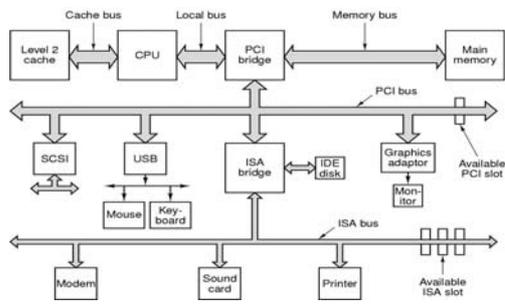
- All'arrivo di una interruzione
  - I registri PC e PSW sono posti sullo *stack* del processo corrente
  - La CPU passa al "modo operativo protetto"
  - Il parametro principale che denota l'interruzione serve come indice nel **vettore delle interruzioni**
    - Così si individua il gestore designato a servire l'interruzione
  - La parte **immediata** del gestore esegue nel contesto del processo interrotto
    - La parte del servizio meno urgente può essere invece **differita** e demandata a un processo dedicato

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 106/121

## Pentium: architettura fisica – 1



Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 107/121

## Pentium: architettura fisica – 2

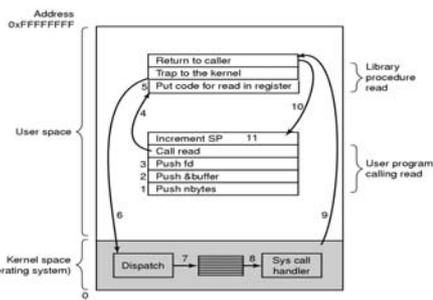
- **PCI bus (Peripheral Component Interconnect 66 MHz, 8 B/ciclo)**
  - Di vecchia concezione ma dotato di connettori per una grande varietà di dispositivi
- **USB bus (Universal Serial Bus, ≤ 1.5 MB/s)**
  - Per l'interconnessione di dispositivi lenti
  - 4 linee delle quali 2 di alimentazione del dispositivo
  - Bus con singolo *master* centrale predefinito che interroga @ 1 ms i dispositivi collegati (*polling!*)
- **SCSI bus (Small Computer System Interface ≤ 160 MB/s)**
  - Per l'interconnessione di dispositivi veloci

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 108/121

## Chiamate di sistema – 1



Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 109/121

## Chiamate di sistema – 2

- La maggior parte dei servizi del S/O sono eseguiti in risposta a invocazioni esplicite di processi
  - Chiamata di sistema
- Le chiamate di sistema sono nascoste in **procedure di libreria** predefinite note ai compilatori
  - L'applicazione **non** effettua direttamente chiamate di sistema
  - La procedura di libreria svolge il lavoro di preparazione necessario ad assicurare la corretta invocazione della chiamata di sistema
- La prima istruzione di una chiamata di sistema (*trap*) deve attivare il **modo operativo privilegiato**
  - Il parametro della chiamata designa l'azione da svolgere e la convenzione per trovare gli altri eventuali parametri
  - Il meccanismo complessivo è simile a quello già visto per il trattamento delle interruzioni
    - Le interruzioni sono **asincrone**
    - Le chiamate di sistema sono **sincrone**

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 110/121

## Chiamate di sistema – 3

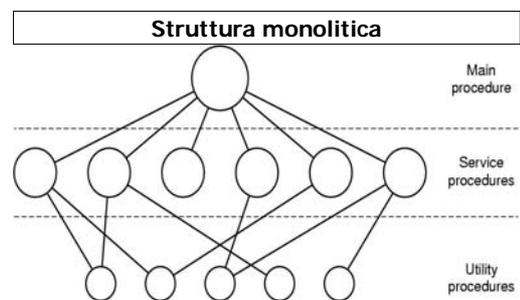
1. L'applicazione effettua una chiamata di sistema
2. Pone sullo *stack* i parametri ...
3. ... secondo una antica convenzione C/UNIX
4. Invoca la procedura di libreria (PL) che corrisponde alla chiamata
5. La PL notifica la chiamata al S/O
  - Scrivendo l'ID della chiamata in un luogo noto al S/O
6. La PL esegue l'istruzione *trap*
  - L'esecuzione prosegue in modo operativo privilegiato
7. Il S/O individua la chiamata da eseguire
8. La esegue
9. Ritorna al chiamante oppure a un nuovo processo
  - Secondo quanto deciso dallo *scheduler*

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 111/121

## Architettura logica di S/O – 1



Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 112/121

## Architettura logica di S/O – 2

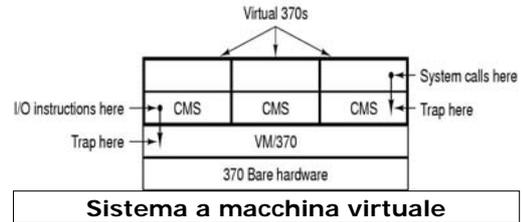
- Un'architettura monolitica **non** ha struttura
  - Il S/O è una collezione "piatta" di procedure
    - Ognuna delle quali può chiamarne qualunque altra
    - Nessuna forma di *information hiding*
  - Il S/O è un singolo .o che collega tutte le procedure che lo compongono
    - Viene incorporato (parzialmente!) negli eseguibili delle applicazioni che lo usano
- L'unica struttura riconoscibile in essa è data dalla convenzione di attivazione delle chiamate di sistema

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 113/121

## Architettura logica di S/O – 3



Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 114/121

## Architettura logica di S/O – 4

- Primi anni '70
  - **VM/370** sviluppato da IBM *Scientific Center*, Cambridge, Massachusetts
  - Basato sull'intuizione che un S/O a divisione di tempo in realtà realizza 2 fondamentali funzioni
    1. Multiprogrammazione
    2. Virtualizzazione dell'elaboratore fisico
  - Separandole e ponendo 2. alla base si possono offrire copie identiche di **macchine virtuali** a S/O distinti che realizzano 1. secondo un criterio loro proprio

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 115/121

## Architettura logica di S/O – 5

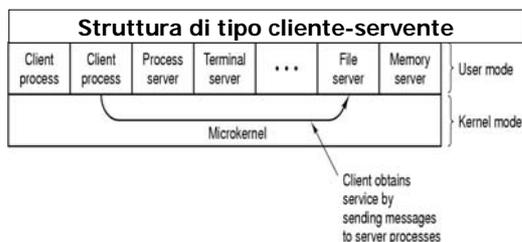
- **CMS (Conversational [Cambridge] Monitor System)**
  - S/O interattivo a divisione di tempo mono-utente
  - Esegue sopra una **macchina virtuale** realizzata da VM/370
- L'idea della virtualizzazione di elaboratori logici o fisici ha avuto notevole seguito
  - Intel: modo 8086 virtuale su Pentium
  - MS Windows & co.: ambiente virtuale di esecuzione MS-DOS
  - JVM: architettura portabile di elaboratore logico

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 116/121

## Architettura logica di S/O – 6



Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 117/121

## Architettura logica di S/O – 7

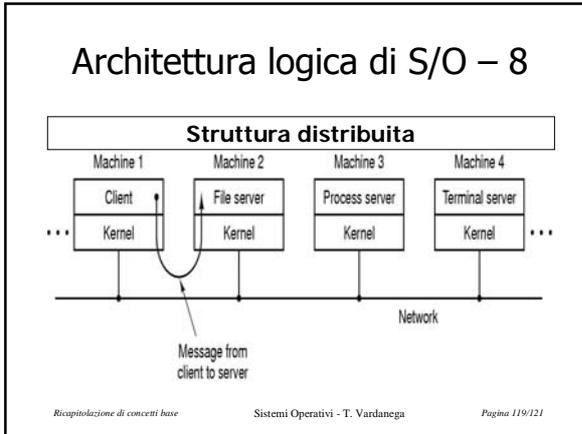
- L'architettura di S/O a modello cliente-servente è anche detta a **micro-kernel**
- L'idea portante è di limitare al solo essenziale le responsabilità del nucleo delegando le altre a processi di sistema **nello spazio di utente**
  - I processi di sistema sono visti come server
  - I processi utenti sono visti come clienti
- Il ruolo del nucleo di S/O è di gestire i processi e supportare le loro comunicazioni
- Idea "pulita" ma prestazioni scadenti
  - Principalmente a causa della grande quantità di memoria che viene copiata da modo nucleo a modo utente

Ricapitolazione di concetti base

Sistemi Operativi - T. Vardanega

Pagina 118/121

## Architettura logica di S/O – 8



## Unità metriche – 1

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
$10^{-3}$	0.001	milli	$10^3$	1,000	Kilo
$10^{-6}$	0.000001	micro	$10^6$	1,000,000	Mega
$10^{-9}$	0.000000001	nano	$10^9$	1,000,000,000	Giga
$10^{-12}$	0.000000000001	pico	$10^{12}$	1,000,000,000,000	Tera
$10^{-15}$	0.000000000000001	femto	$10^{15}$	1,000,000,000,000,000	Peta
$10^{-18}$	0.000000000000000001	atto	$10^{18}$	1,000,000,000,000,000,000	Era
$10^{-21}$	0.000000000000000000001	zepto	$10^{21}$	1,000,000,000,000,000,000,000	Zetta
$10^{-24}$	0.00000000000000000000001	yocto	$10^{24}$	1,000,000,000,000,000,000,000,000	Yotta

Ricapitolazione di concetti base      Sistemi Operativi - T. Vardanega      Pagina 120/121

## Unità metriche – 2

**Prefixes for binary multiples**

In December 1998 the International Electrotechnical Commission (IEC), the leading international organization for worldwide standardization in electrotechnology, approved six new IEC international standard names and symbols for prefixes for binary multiples for use in the fields of data processing and data transmission. The prefixes are as follows:

Factor	Name	Symbol	Origin	Derivation
$2^{10}$	kibi	Ki	kibibinary: $(2^{10})^1$	kibi: $(10^3)^1$
$2^{20}$	mebi	Mi	megabinary: $(2^{10})^2$	mega: $(10^3)^2$
$2^{30}$	gibi	Gi	gigabinary: $(2^{10})^3$	giga: $(10^3)^3$
$2^{40}$	tebi	Ti	terabinary: $(2^{10})^4$	tera: $(10^3)^4$
$2^{50}$	pebi	Pi	petabinary: $(2^{10})^5$	peta: $(10^3)^5$
$2^{60}$	exbi	Ei	exabinary: $(2^{10})^6$	exa: $(10^3)^6$

**Examples and comparisons with SI prefixes**

one **kibibit** 1 Kibit =  $2^{10}$  bit = **1024 bit**  
 one **kibibyte** 1 KiB =  $10^3$  bit = **1000 bit**  
 one **mebibyte** 1 MiB =  $2^{20}$  B = **1 048 576 B**  
 one **megabyte** 1 MB =  $10^6$  B = **1 000 000 B**  
 one **gibibyte** 1 GiB =  $2^{30}$  B = **1 073 741 824 B**  
 one **gigabyte** 1 GB =  $10^9$  B = **1 000 000 000 B**

It is suggested that in English, the first syllable of the name of the binary multiple given should be pronounced in the same way as the first syllable of the name of the corresponding SI prefix, and that the second syllable should be pronounced as "base".

It is important to recognize that the new prefixes for binary multiples are not part of the International System of Units (SI), the modern metric system.

Ricapitolazione di concetti base      Sistemi Operativi - T. Vardanega      Pagina 121/121