

La piattaforma Microsoft Windows: inquadramento storico

Windows nasce nel 1985. Prime due versioni hanno forti limitazioni. Girano in modalità reale 8086 (limite a 640k di memoria) o in modalità protetta 80286.

Prima versione “utilizzabile” (Windows 3.0) del 1991 e sfrutta le capacità dei processori 386. Nascono le prime applicazioni commerciali di buon livello.

Anni 80, uso di cosiddetti pacchetti integrati (Symphony, Framework...): unica applicazione con tutte le funzioni più importanti (word processor, foglio elettronico, database, comunicazioni). Questi pacchetti squilibrati: una delle funzioni completa, altre insoddisfacenti.

Obiettivi di Windows:

- ambiente grafico di interfaccia con utente
- spostare integrazione tra funzioni applicative a livello di ambiente operativo. Utente può quindi scegliere applicativi specializzati migliori lasciando al sistema operativo il compito di integrarli.

La piattaforma Windows -- 1

La piattaforma Microsoft Windows

Si tratta di una famiglia di ambienti, indicati di solito come **WinXX**.

4 sistemi: Windows 3.x (16 bit), Windows 95/98 (ibrido 16/32), Windows NT (32 bit) e Windows CE (32bit).

- **Windows 3.x** è in fase di abbandono ma ha ancora una notevole base di installato, soprattutto nelle grandi aziende.
- **Windows 95/98** è destinato al mercato “consumer” e a quello dei portatili. È un ambiente ibrido in cui convivono parti a 16 bit e parti a 32 bit, orientato a fornire la massima compatibilità con il DOS, con problemi intrinseci di fragilità.
- **Windows NT** è stato pensato come sistema per uso professionale. È un 32bit vero con un’architettura microkernel molto robusta (basato in parte su tecnologia Digital VMS) che si rivolge allo stesso mercato dei sistemi Unix (comprende un sottosistema conforme a Posix).
- **Windows CE** è destinato al mercato palmari ed embedded.

NT5 come principale piattaforma Microsoft per il futuro (Windows 2000):
laptop, desktop, server, mono/multi-processori

La piattaforma Windows -- 2

La programmazione in Windows

Funzioni del sistema operativo invocate tramite **API** (Application Program Interface), interfaccia fra applicazioni e SO.

Esistono 2 API Windows: **Win16** (per i sistemi a 16 bit) e **Win32** (per i sistemi a 32 bit). A parte alcune piccole differenze Win95/98 e WinNT hanno la stessa API.

API Windows implementate da librerie ad aggancio dinamico (**DLL**).

Struttura modulare sistema operativo. API principali fornite da 3 DLL:

- **KERNEL32.dll**; implementa le funzioni per gestione memoria, thread e processi
- **USER32.dll**; interfaccia con utente e sistema di finestre
- **GDI32.dll**; Graphics Device Interface, funzioni di grafica

Estensibilità: API si espande aggiungendo nuove DLL (per esempio WinSocket per la comunicazione su TCP/IP)

Windows 95/98 e NT

Windows 95/98

Windows NT (versioni workstation e server)

Vantaggi NT rispetto a 95/98

- Supporto ai sistemi multiprocessore (fino a 8 CPU)
- NT portabile, multiplatforma (es. DEC alpha, PowerPC, Intel),
95/98 solo su Intel x86
- NT multiutente (autenticazione e controllo accesso ai file)
- NT è un sistema operativo a 32 bit
- 95 ha molte parti di codice non rientranti (Win16), da eseguire in mutua esclusione (possibile blocco sistema)
- NT ha protezione address space (diversi processi non possono interferire)

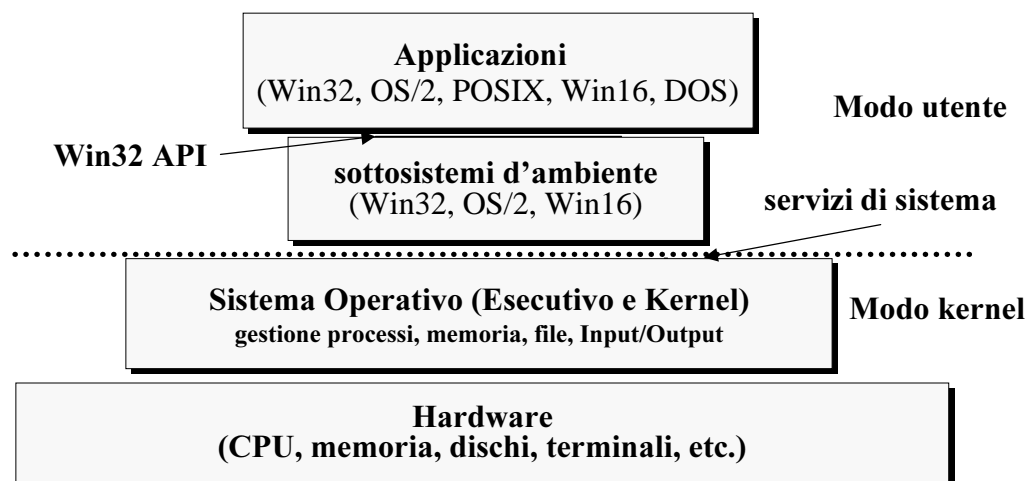
Windows NT (New Technology)

Obiettivi di progettazione

- **Portabilità** (scritto in C e C++)
- **Scalabilità** (per funzionare su diverse piattaforme, anche multiprocessori)
- Funzionalità di **networking**
- **Interoperabilità** (capacità di interagire con altri S.O.)
- **Estendibilità** (struttura stratificata e modulare, per facilitare innovazione)
- **Sicuro** (classe C2: passwd-login, protezione memoria e S.O., quote)
- **Affidabilità** (protezione Hardware memoria virtuale, file system NTFS)
- **POSIX** compatibile (a livello sorgente)

Windows NT -- 22

Architettura di Windows NT

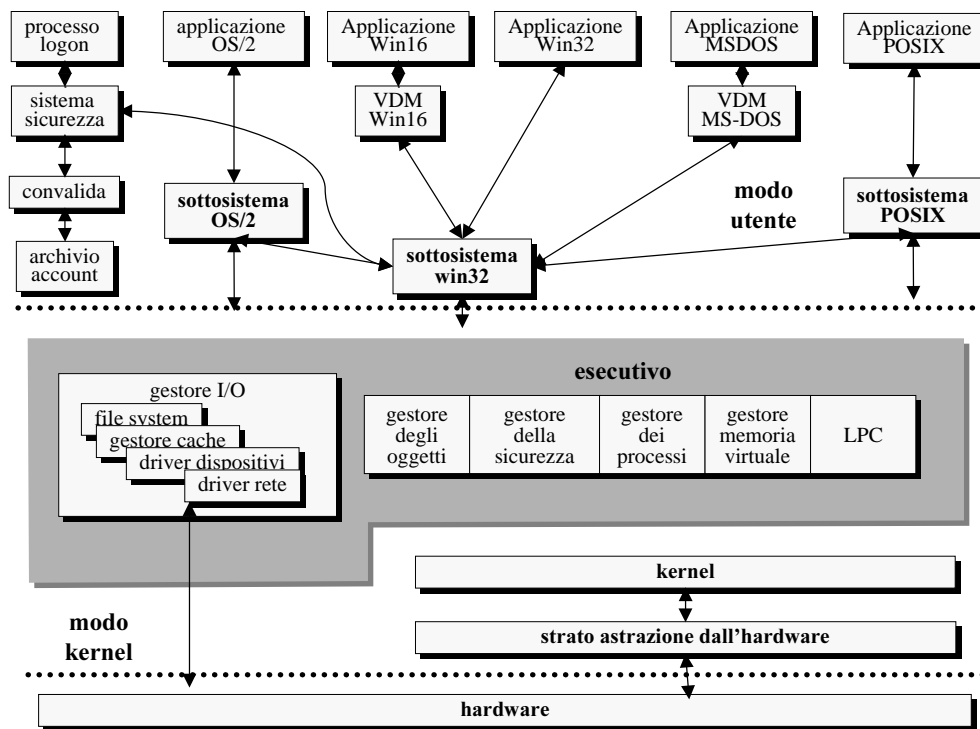


Win32 API sono funzioni (documentate) di interfaccia con il sottosistema di ambiente

I **servizi di sistema** sono funzioni (non documentate) d'interfaccia con l'esecutivo

Windows NT -- 23

Architettura di Windows NT



Windows NT -- 24

Principi di progettazione di NT

Architettura a **microkernel**: molti dei servizi del sistema operativo sono realizzati nello spazio utente (modello Client/Server)

Tutte le risorse del sistema sono rappresentate in termini di oggetti (oggetto processi, oggetto thread, oggetto file, oggetto memoria)

Processi usano oggetti tramite descrittori

Funzioni per la gestione degli oggetti

Progettato per il multiprocessing simmetrico

Windows NT -- 25

Strato di astrazione dell'hardware

L'Hardware Abstraction Layer:

- nasconde le differenze tra le diverse piattaforme hardware su cui può eseguire Windows NT
- realizza una macchina virtuale usata come interfaccia dagli altri moduli (kernel, esecutivo e anche driver dispositivi)

Kernel

Architettura a **microkernel**: molti servizi sono realizzati nello spazio utente (modello Client/Server)

Il kernel di Windows NT è non preemptive ed è incaricato di:

- scheduling dei thread
- gestione interrupt ed eccezioni

Il kernel opera su oggetti (oggetto thread, oggetto processo, oggetto semaforo etc.)

Oggetto **processo** è dotato di spazio di indirizzamento e caratterizzato da alcune informazioni (utente, priorità, etc.)

Ogni oggetto processo può avere più oggetti **thread** (flussi di esecuzione all'interno di uno stesso spazio di indirizzamento)

Tutte risorse implementate in termini di **oggetti** (processi, thread, file, memoria)

Lo scheduling in Windows NT

Unità di scheduling in NT sono i **thread**

NT ha uno **scheduling preemptive** con diversi livelli di priorità

Scheduling preemptive, con 32 livelli di priorità suddivisi in tre classi:

- classe **real time** (16-31)
- classe a **priorità variabile** (1-15)
- classe di **sistema**

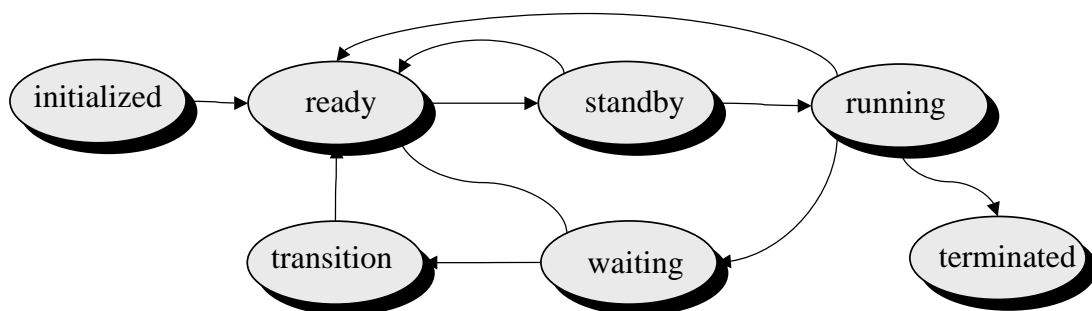


Una coda per ogni livello di priorità con **round robin** in ogni coda.

In ogni istante il kernel esegue il thread (di qualunque classe) con priorità più alta, **preemption** dei thread con priorità più bassa (messi in testa a coda ready)

La priorità di un thread viene definita in relazione alla priorità del processo che lo ospita, ma i thread della **classe a priorità variabile** possono cambiarla dinamicamente (per es., NT abbassa la priorità dei thread CPU bound e privilegia i thread interattivi e I/O bound)

Scheduling dei thread

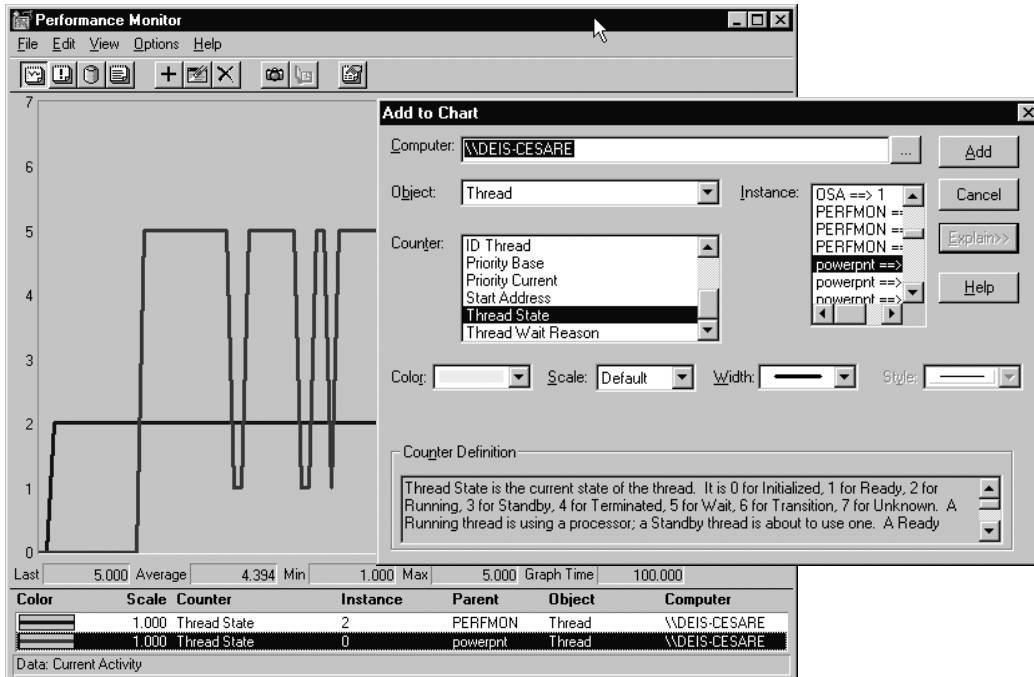


Caso **multiprocessore**:

- default, i thread vanno su qualunque processore
- processor affinity, si può legare un thread a un certo processore
- uno stato di esecuzione e uno stato di standby per ogni processore

Esempio: visualizzazione stato dei thread

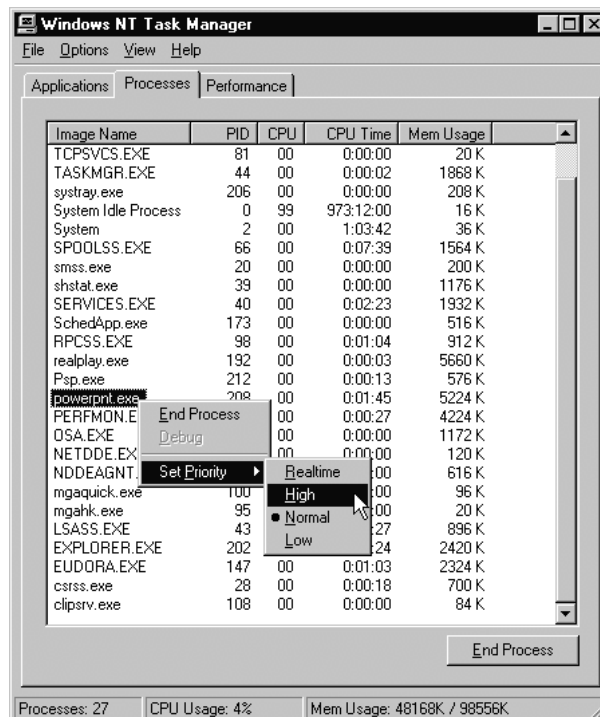
Si può utilizzare il Performance Monitor (presente tra gli administrative tools) per visualizzare gli stati in cui si trovano i thread presenti su una macchina.



Windows NT -- 30

Esempio: cambio priorità thread

Si può utilizzare il Task Manager (ctrl+shift+esc) per vedere lo stato dei processi, dei thread e per cambiare la priorità



Windows NT -- 31

La starvation in Windows NT

Un thread sempre in esecuzione può impedire l'accesso alla CPU dei thread con priorità inferiore, causando la starvation di tali thread.

Il “balance set manager” è un thread di sistema di NT che periodicamente controlla se ci sono thread ready che non riescono ad acquisire la CPU, nel qual caso aumenta la priorità di tali thread al massimo (15) per un certo periodo di tempo.

Questo meccanismo agisce solo sui thread normali, non real time.

Interrupt ed eccezioni

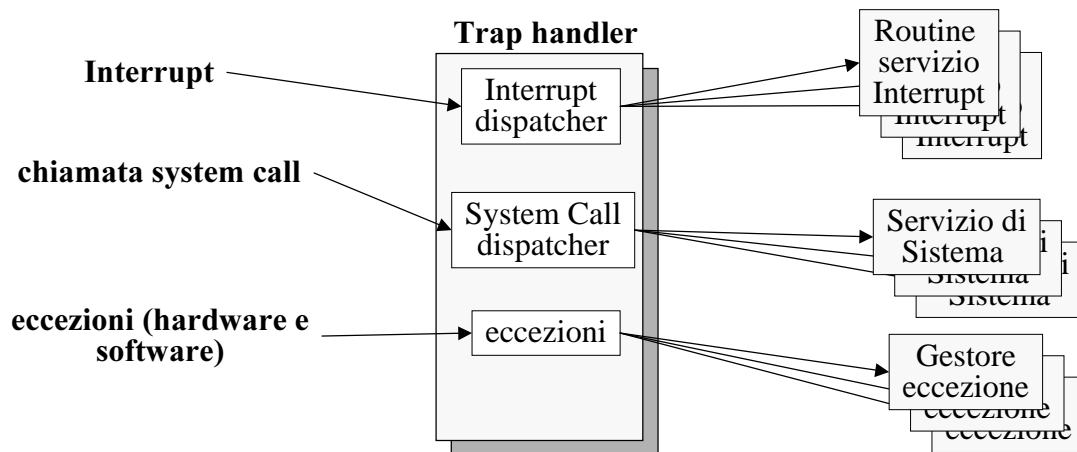
- 1) interrupt sono eventi asincroni
- 2) eccezioni sono eventi sincroni
- 3) system call richiedono servizio al sistema operativo

1, 2, 3 sono gestiti tutti nello stesso modo, sono intercettati da un unico trap handler che li smista al corrispondente gestore

Livelli di interrupt (32) mascherabili indipendentemente per ogni processore

Definizione di un insieme di interrupt indipendenti dall'hardware.

Gestione Interrupt ed eccezioni



- un device driver fornisce una routine di servizio a un interrupt proveniente da un dispositivo (**interrupt hardware**)
- la primitiva kill invia un **interrupt software** a un processo che esegue una opportuna routine di gestione
- una divisione per zero scatena una **eccezione software**
- un errore di parità della memoria genera una **eccezione hardware**
- una **chiamata di sistema** (es., read()) richiede un servizio al SO, che esegue il codice corrispondente in vece del processo richiedente

Windows NT -- 34

Interrupt ed eccezioni

Kernel disabilita interruzioni nell'esecuzione di sezioni critiche

NT ha 32 livelli di interrupt (IRQL) mascherabili indipendentemente per ogni processore

IRQL hanno priorità, interrupt serviti rispettando priorità:

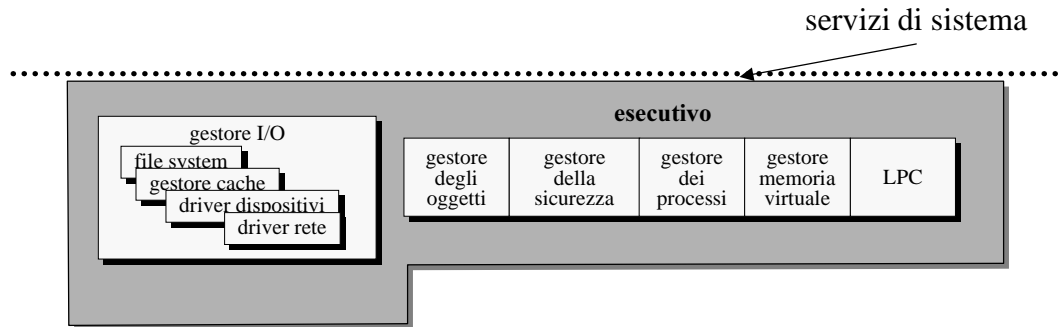
- priorità maggiori possono fare preemption di gestione interrupt minori
- servizio di interrupt con priorità minori viene bloccato durante lo svolgimento servizio interrupt priorità maggiore

NT definisce un insieme di interrupt indipendenti dall'hardware.

Windows NT -- 35

Modulo esecutivo

Il modulo esecutivo di Windows NT fornisce una serie di servizi utilizzabili da tutti i sottosistemi di ambiente



Gestore oggetti

NT rappresenta tutte le risorse in termini di **oggetti**.

Esempi di oggetti: directory, file, semafori, eventi, thread, porte

Il gestore degli oggetti viene invocato quando si deve gestire uno qualunque degli oggetti NT.

Esempio: un processo che vuole accedere a un file, esegue una open che porta il gestore degli oggetti a restituire al processo un descrittore (un handle) per l'oggetto file richiesto.

Ogni processo ha una tabella degli oggetti riferiti (come tabella file descriptor di Unix)

Gestore della memoria virtuale

Requisito NT: piattaforma hardware deve avere un **meccanismo di paginazione**

Indirizzi a 32 bit: spazio indirizzamento virtuale di 4 GB per processo (2 per il processo e 2 per il sistema operativo)

Pagine di 4 KB e tabella delle pagine a due livelli (10 bit address per ogni livello)

Strategia sostituzione pagine di tipo **FIFO per ogni processo** (modifica dinamica del working set per ogni processo a seconda delle esigenze mostrate, cioè del numero di page fault)

Anche memoria rappresentata tramite **oggetti**:

- diversi processi possono **condividere oggetti di memoria** (condivisione handle oggetti oppure con speciali oggetti di sezione)
- oggetti di memoria condivisi mappati in pagine fisiche condivise (problemi di consistenza, uso di un livello in più di indirettezza.)

Il meccanismo di **swap** si appoggia sul file di paginazione su disco.

Gestore dei processi

Offre i servizi per la creazione, l'eliminazione e la gestione di thread e processi

è il gestore di basso livello => non possiede informazioni sulle relazioni e gerarchie tra processi, informazioni mantenute nel sottosistema d'ambiente relativo al processo

Interagisce con il gestore degli oggetti, poiché processi e thread sono oggetti

La sincronizzazione dei thread

Sincronizzazione nel kernel:

- il kernel porta IRQL al massimo livello nel caso monoprocesore
- il kernel usa spinlock in memoria condivisa per assicurare la mutua esclusione di diversi processori (attesa attiva)

Costrutti di sincronizzazione di più alto livello:

- i thread possono essere sincronizzati su molti oggetti (processi, thread, file, console input, mutexes, semafori, eventi, timers)
- un thread che si sospende su un oggetto (invocando la API win32 *WaitForSingleObject()*) viene risvegliato quando l'oggetto entra in uno stato *signaled*

Uso di mutexes per sincronizzare diversi thread (anche di diversi processi)

Semafori (identificati da stringa alfanumerica) sono in stato *signaled* quando il loro valore è maggiore di 0

Windows NT -- 41

Chiamata di procedura locale (LPC)

La chiamata di procedura **locale** (Local Procedure Call) è il meccanismo di comunicazione a scambio di messaggi tra processi client e server **appartenenti al sistema operativo** e residenti su una stessa macchina

Ogni processo **server** rende noto un oggetto “**porta** di connessione”

Un processo **client** può aprire un oggetto porta, ottenendo in risposta un oggetto canale di comunicazione su cui può avvenire lo scambio di messaggi tra client e server.

NT ha diverse tecniche di scambio di messaggi, adatte a messaggi di lunghezza diversa

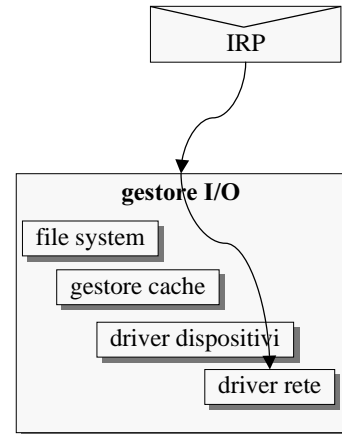
Windows NT -- 42

Il gestore di I/O

Gestione di:

- file system
- cache
- dispositivi
- rete

gestore I/O riceve pacchetti standardizzati di richiesta di I/O (IRP, I/O Request Packet) che smista al corrispondente driver.



Windows NT -- 43

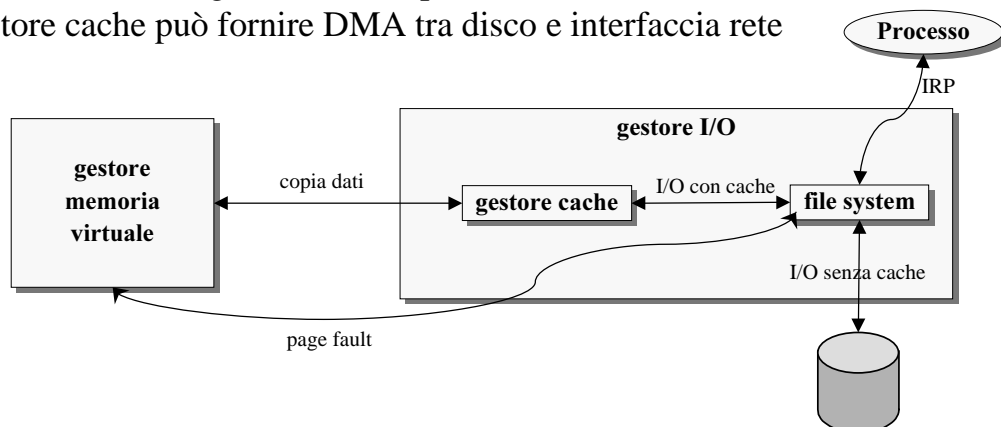
Il gestore della cache

NT ha un gestore della cache, non è quindi il file system a occuparsene.

NT riserva per caching fino a 1 GB dello spazio di indirizzamento di 2 GB riservato al SO dal gestore della memoria virtuale. Strategie gestione cache:

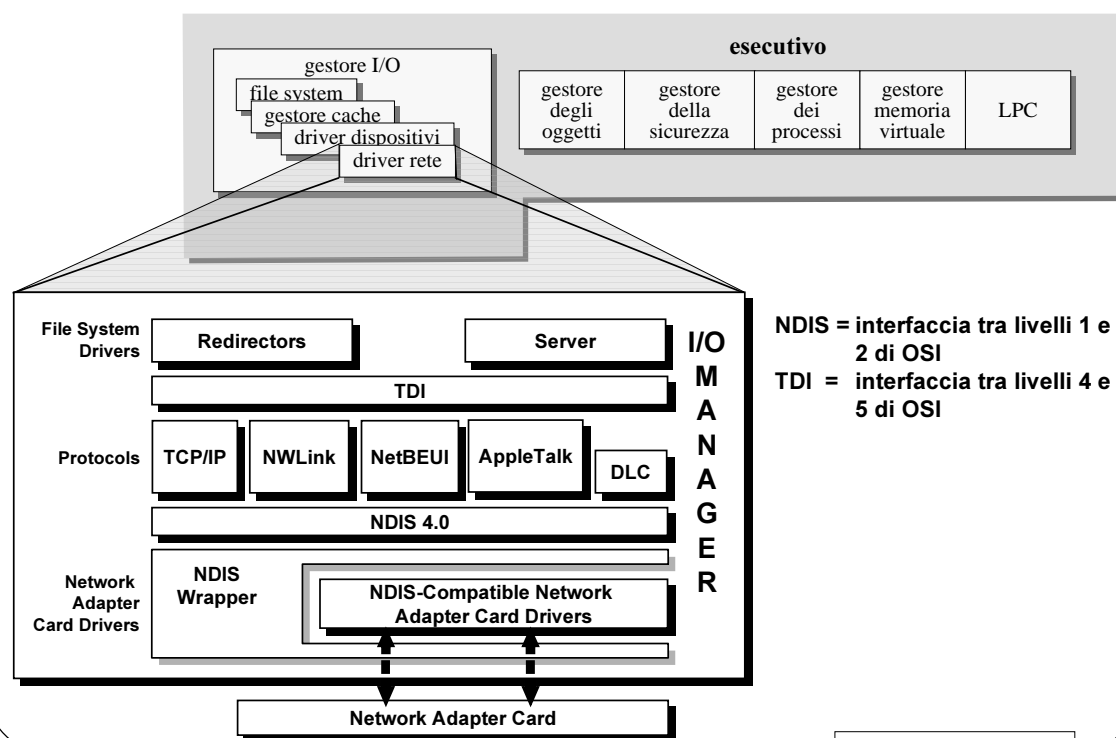
- write-back (ogni 4-5 secondi)
- write-through (a richiesta processo)

gestore cache può fornire DMA tra disco e interfaccia rete



Windows NT -- 44

Il driver di rete



Windows NT -- 45

Programmazione di rete in Windows NT

Pipe con nome: simile alle FIFO Unix, nome presente nel file system, controlli di sicurezza come per i file, comunicazione processi su una stessa macchina o su macchine differenti (diverso da Unix)

Caselle postali (mailslot): scambio di messaggi senza connessione non affidabile

Socket Windows: la piattaforma Windows ha una API (Winsock) che offre le funzioni per l'uso delle socket (interfaccia al TCP/IP). La Winsock API è implementata come DLL.

RPC: chiamate di procedura remota (conforme standard OSF DCE), con marshalling parametri, Microsoft IDL (Interface Definition Language) e relativo compilatore di protocollo per generazione stub

Windows NT -- 46

La sicurezza in Windows NT

NT 3.51 certificato di classe C2 (livello Orange Book):

- autenticazione via password
- Discretionary Access Control (DAC), ogni oggetto ha una Access Control List che specifica i tipi di accesso da parte di utenti e gruppi
- security auditing, NT rileva e registra gli eventi ritenuti importanti ai fini della sicurezza
- protezione della memoria, ogni processo accede SOLO alla propria memoria virtuale (e le pagine di memoria sono cancellate prima di essere allocate a un processo)

Gestore sicurezza

Il gestore della sicurezza (Security Reference Monitor, SRM) è un modulo dell'esecutivo che controlla e impone la politica di sicurezza.

Verifica e controlla se un processo utente è autorizzato ad accedere o a utilizzare un qualunque oggetto del sistema (file, dispositivi, processi, finestre, servizi, etc.)

Il gestore della sicurezza confronta l'**oggetto di accesso** associato a ogni processo con la **Access Control List** (ACL) dell'oggetto che il processo vuole usare

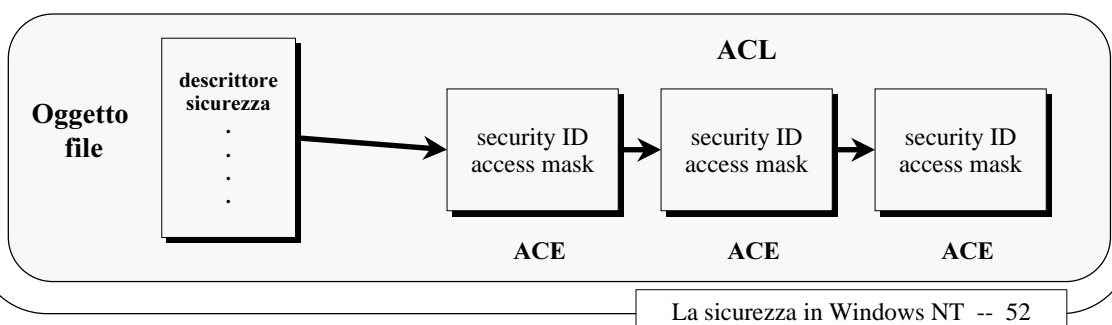
Per motivi di efficienza, autorizzazione eseguita SOLO al momento dell'apertura dell'oggetto (i processi di kernel usano puntatori a oggetti, non handle e quindi bypassano l'autorizzazione)

La protezione degli oggetti NT

Oggetti di NT che possono essere protetti: file, directory, dispositivi, pipe, processi, thread, eventi, mutex e semafori, timer, oggetti d'accesso per utenti, finestre, desktop, registry, stampanti.

Ogni oggetto ha un **descrittore di sicurezza** contenente:

- l'ID di sicurezza del proprietario
- lista di controllo dell'accesso (ACL), definisce **chi** ha accesso all'oggetto e **cosa** può fare; lista di Access Control Entry (ACE), ogni ACE permette o vieta l'accesso all'oggetto da parte di un security ID
- lista di controllo dell'accesso di sistema (SACL), quali operazioni eseguite da quali utenti debbano essere registrate e memorizzate



Sottosistemi di ambiente e Applicazioni

I sottosistemi di ambiente sono processi eseguiti in **modo utente** e utilizzano i servizi dell'esecutivo (i servizi di sistema).

Ogni sottosistema fornisce una **API** alle applicazioni.

Interfaccia di programmazione principale: **Win32**

Altri sottosistemi: **POSIX, OS/2**

Applicazioni Win16 e DOS eseguono su una macchina virtuale al di sopra del sottosistema d'ambiente Win32

Sottosistema di **accesso e di sicurezza**: il sottosistema di accesso autentica un utente tramite password, il sottosistema di sicurezza genera un oggetto di accesso per ogni utente autenticato, contenente id utente, privilegi e quote. Oggetto di accesso controllato dal gestore di sicurezza ogni volta l'utente tenta di accedere a un oggetto del sistema.



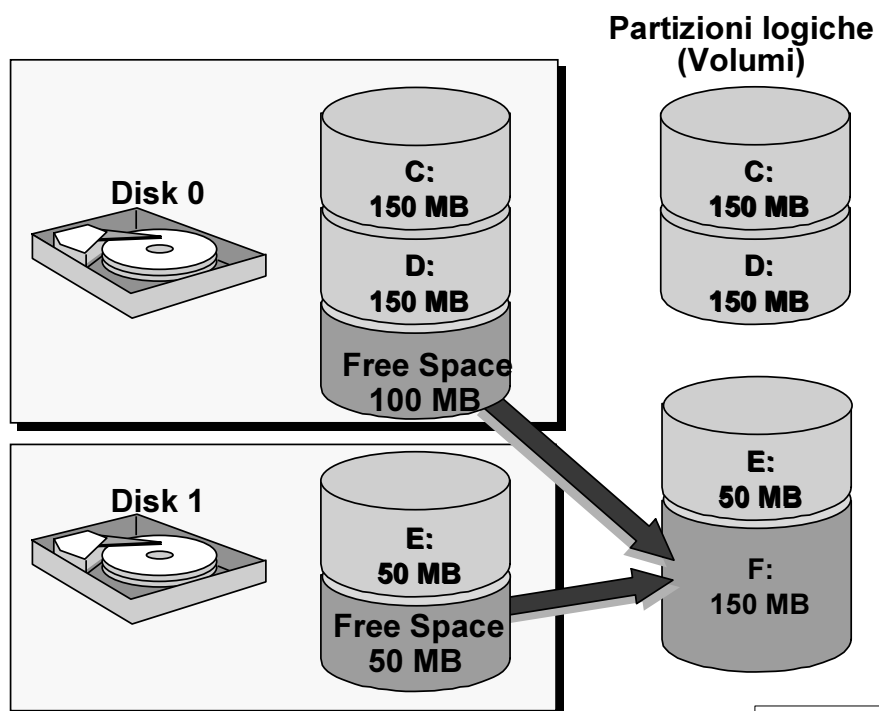
File System

Windows NT supporta molti file system: FAT, NTFS, HPFS

Il file system specifico di NT è **NTFS**, che presenta alcune caratteristiche:

- riduzione della frammentazione interna (migliore uso spazio rispetto FAT)
- tolleranza ai guasti
- sicurezza
- supporto a file e file system di grandi dimensioni
- compressione dati

Organizzazione fisica file system NTFS: i Volumi



Organizzazione fisica file system NTFS

Unità di allocazione (cluster) di pochi KB (4 KB per dischi > 2 GB):

- bassa frammentazione interna

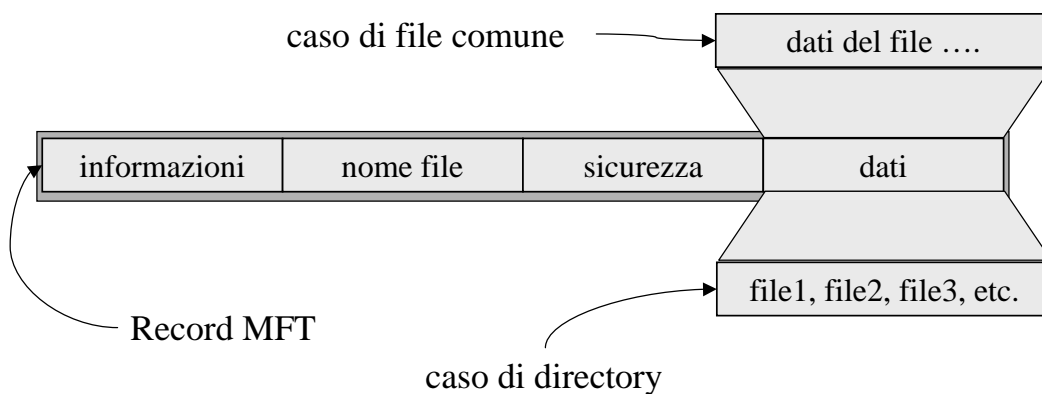
Un file NTFS è un oggetto, costituito di attributi:

- nome del file
- data e ora di creazione
- ACL (lista controllo accessi)
- dati contenuti nel file (anche i dati sono inseriti in un campo attributo)

Organizzazione fisica file system NTFS

Ogni volume NTFS è strutturato in una MFT (Master File Table) che contiene dei record di 1 KB, che contengono tutti i dati dei file.

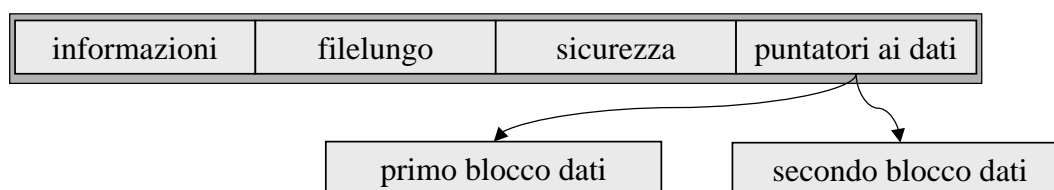
Ogni record MFT descrive un file, contiene infatti tutte le informazioni e i dati del file.



Organizzazione fisica file system NTFS

In particolare, gli attributi di un file si suddividono in:

- *attributi residenti* se sono memorizzati all'interno di un record MFT (es. nome file o anche i dati se di piccole dimensioni)
- *attributi non residenti* se sono memorizzati in estensioni contigue su disco e in questo caso il record MFT contiene un puntatore a tali estensioni

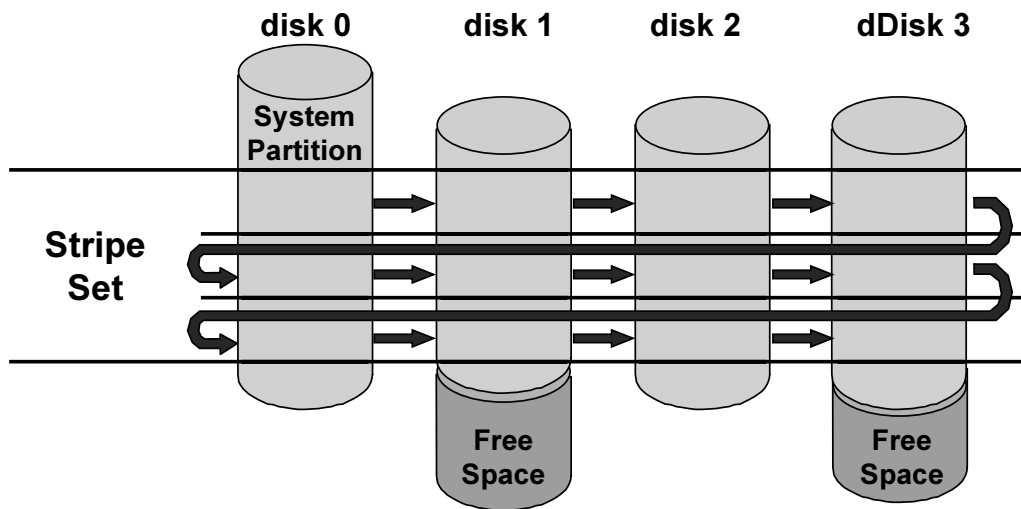


file con molti attributi o di grandi dimensioni memorizzati in molte estensioni possono richiedere più record MFT

NTFS: tolleranza ai guasti e livelli di RAID

RAID 0	Striping
RAID 1	mirroring
RAID 2	Disk striping with error-correction code (ECC)
RAID 3	Disk striping with ECC stored as parity
RAID 4	Disk striping large blocks; parity stored on one drive
RAID 5	Disk striping with parity distributed across multiple drives

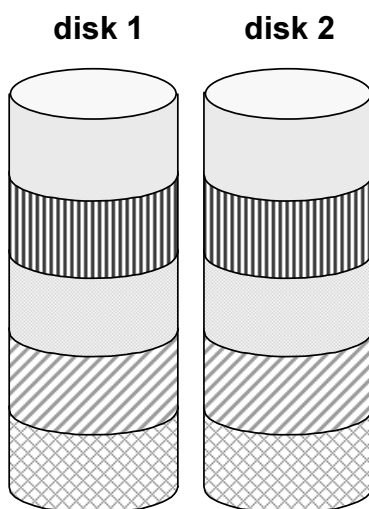
NTFS: RAID livello 0 (striping)



Si crea **un solo volume** logico su tutti i dischi. I dati sono allocati su dischi diversi, per **parallelizzare** operazioni di I/O.

Windows NT -- 61

NTFS: RAID livello 1 (mirroring)



Tutti i dati sono **replicati sui due dischi**. Il sistema scrive un dato sempre su due dischi.

Lettura puo' essere parallelizzata sui due dischi

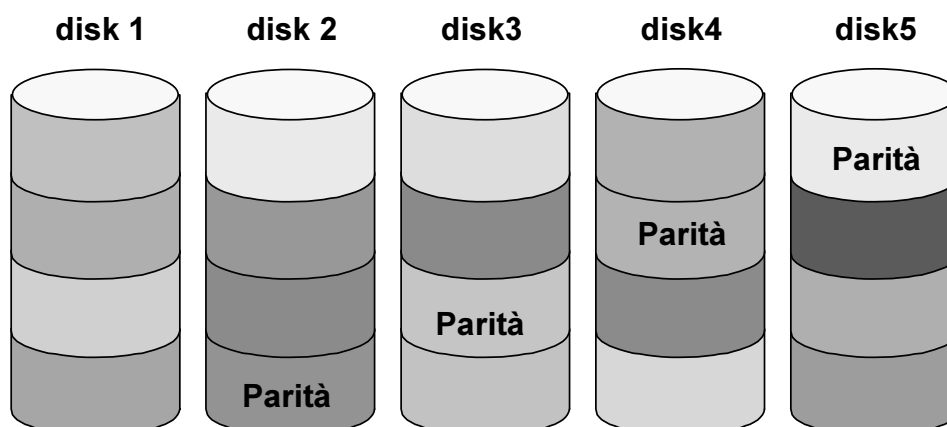
Possibile mirroring anche aree sistema

tolleranza al guasto di un disco

Elevato **costo** (utilizzo dischi del 50%)

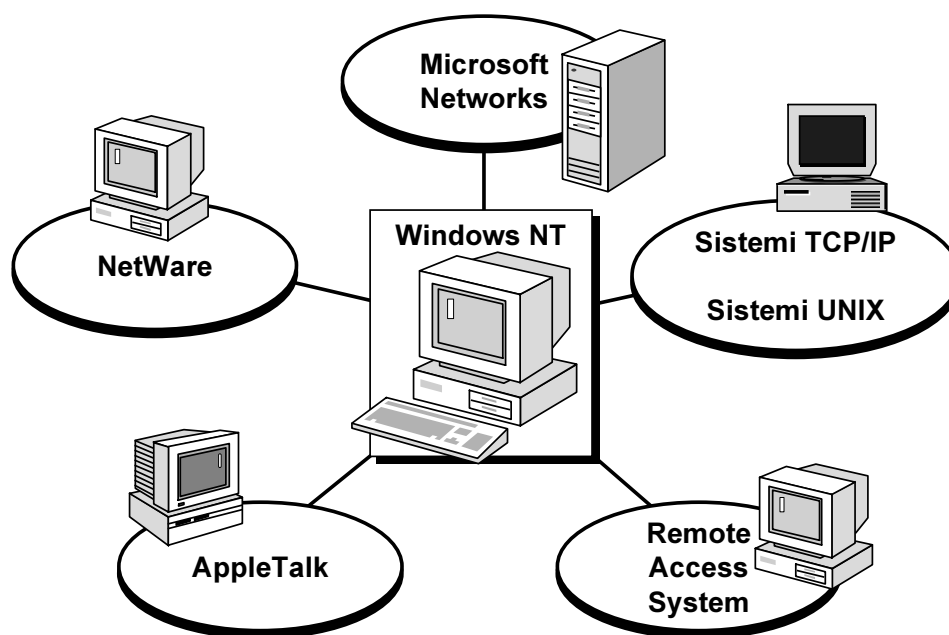
Windows NT -- 62

NTFS: RAID livello 5 (striping con parità)

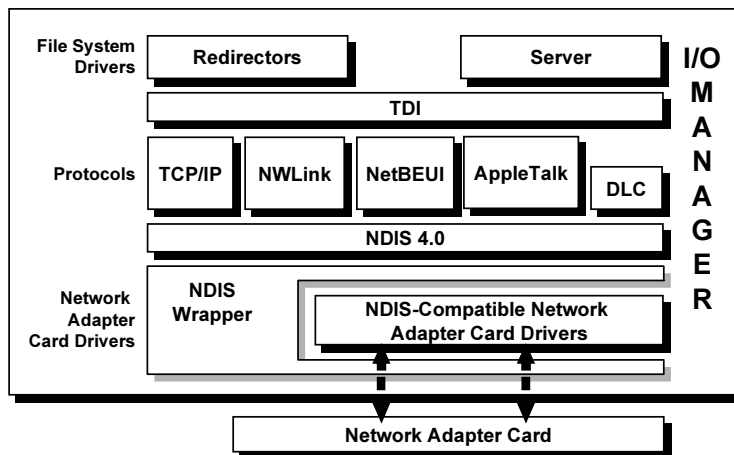


Ogni sezione di parità contiene l'**or-esclusivo** delle 4 sezioni dati corrispondenti. Nel caso di perdita di **UNA** delle sezioni dati, il sistema ricostruisce la perdita utilizzando la sezione di parità. Minore costo rispetto a mirroring (in questo esempio, costo del 20%) Ogni scrittura richiede modifica sezione di parità.

Il supporto alla programmazione di rete



Protocolli di rete supportati



NDIS = interfaccia tra livelli 1 e 2 di OSI

TDI = interfaccia tra livelli 4 e 5 di OSI

TCP/IP = protocollo standard Internet (livelli rete e trasporto OSI)

NetBEUI (NetBIOS Extended User Interface) = costruito sopra al NetBIOS, è usato per definire dei nomi logici per le macchine e per trasmissioni affidabili di dati. Nessuna tecnica di routing dei messaggi, nome logico del computer usato come indirizzo

NWLink = protocollo per connettere client NT a un server NetWare

DLC (Data Link Control) = collegamento a mainframe IBM e stampanti HP

Windows NT -- 65

Gestione di sistemi complessi

Sistemi informativi delle organizzazioni (industrie e aziende) sono sistemi distribuiti **aperti e globali, interconnessi da Internet:**

Problemi:

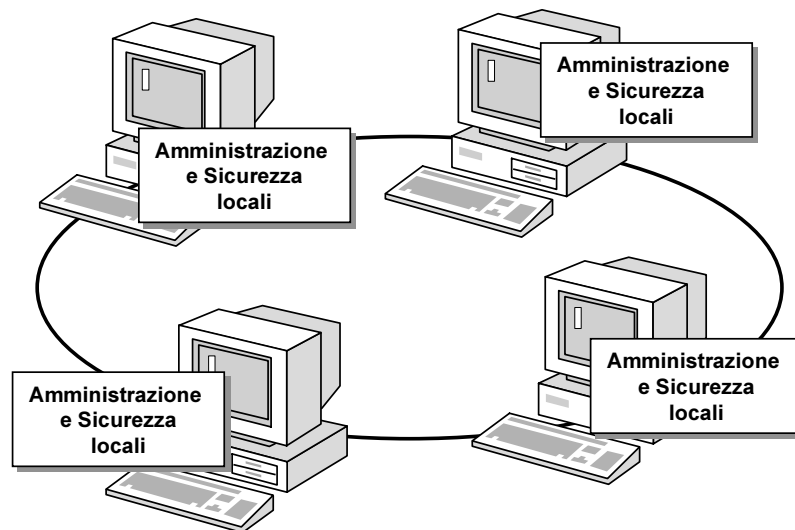
- **eterogeneità** delle piattaforme (varie versioni Windows, Unix, Mac)
- interconnessione di macchine su **scala globale**
- Internet come rete **non sicura**

NT offre vari **modelli di organizzazione** di una rete di calcolatori, offre alcuni servizi multipiattaforma, alcune operazioni sicure

Modelli di organizzazione basati sui **Domini**, intesi come gruppo di stazioni di lavoro e di Server NT che condividono una politica di sicurezza e il database degli utenti

Windows NT -- 66

Il Modello Workgroup



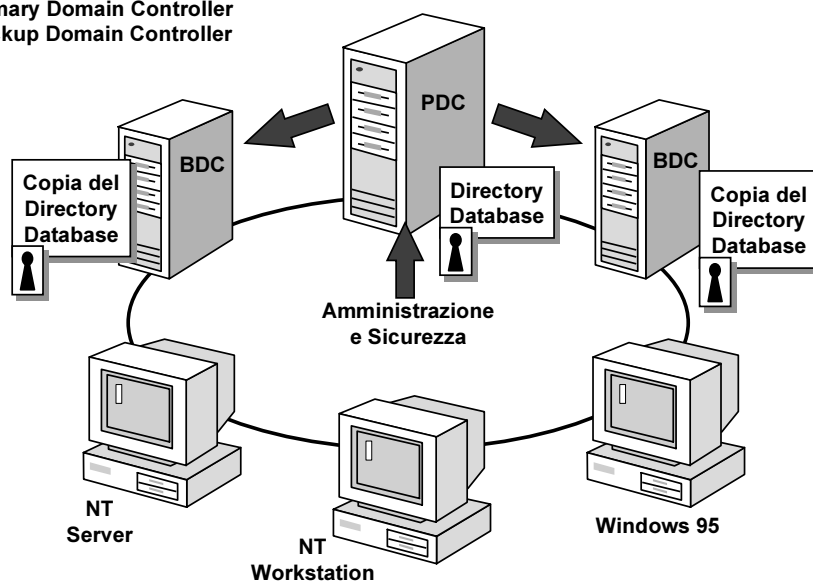
Ogni macchina è gestita autonomamente:

- difficile imposizione politiche globali
- alti costi di gestione

Windows NT -- 67

Il Modello a Dominio

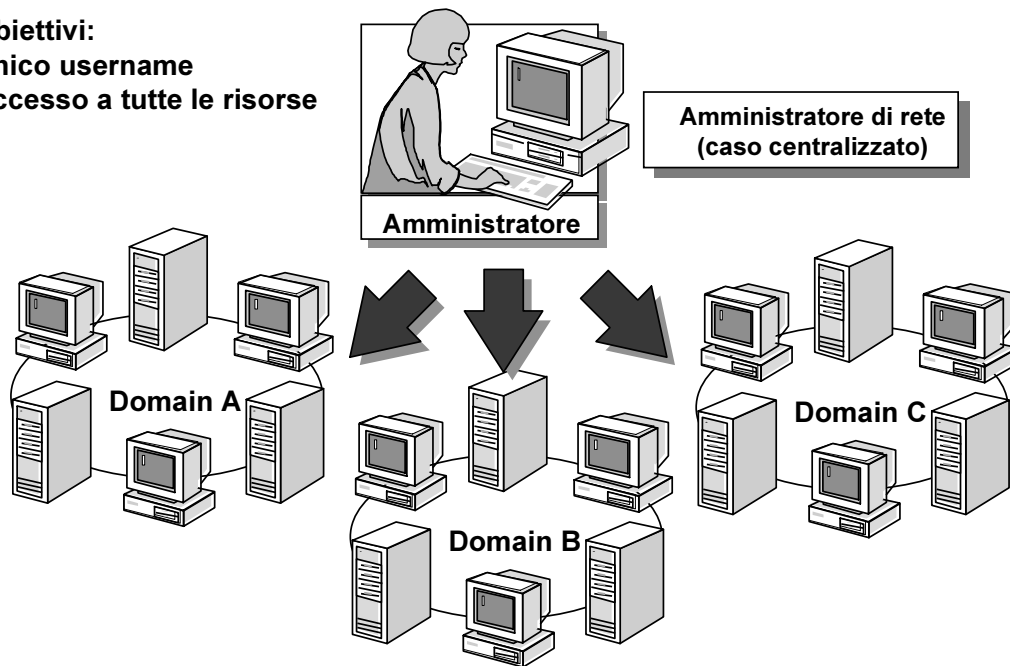
PDC = Primary Domain Controller
BDC = Backup Domain Controller



Windows NT -- 68

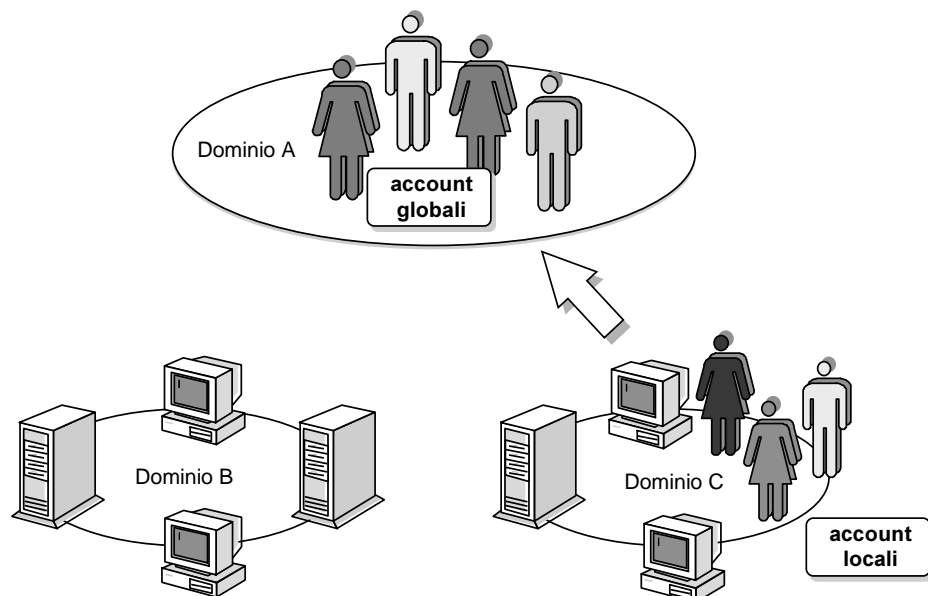
Gestione di reti complesse

Obiettivi:
Unico username
Accesso a tutte le risorse



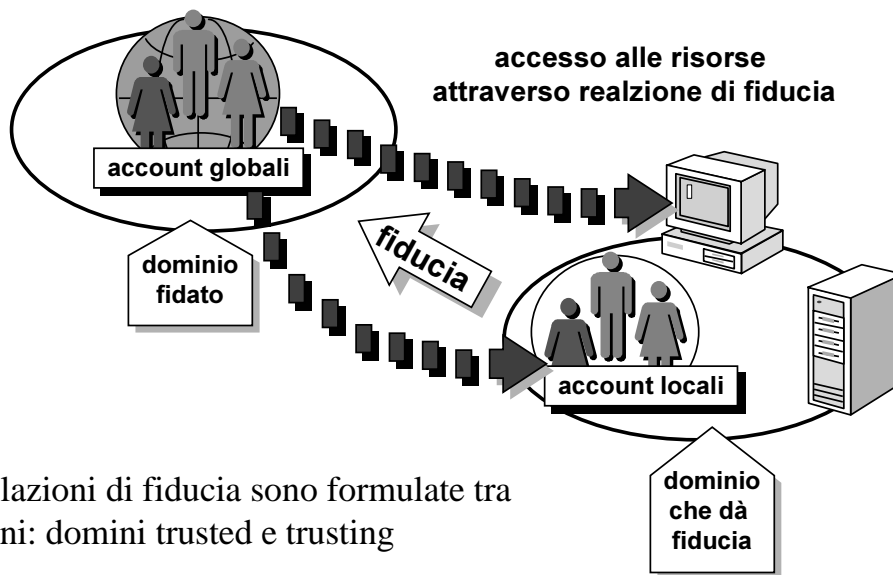
Windows NT -- 69

Tipi di account gestiti in NT



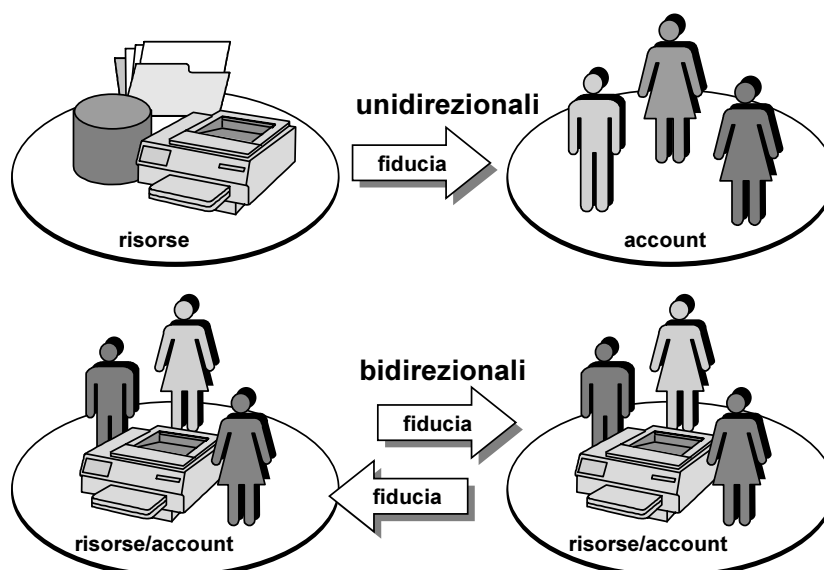
Windows NT -- 70

Relazioni di fiducia

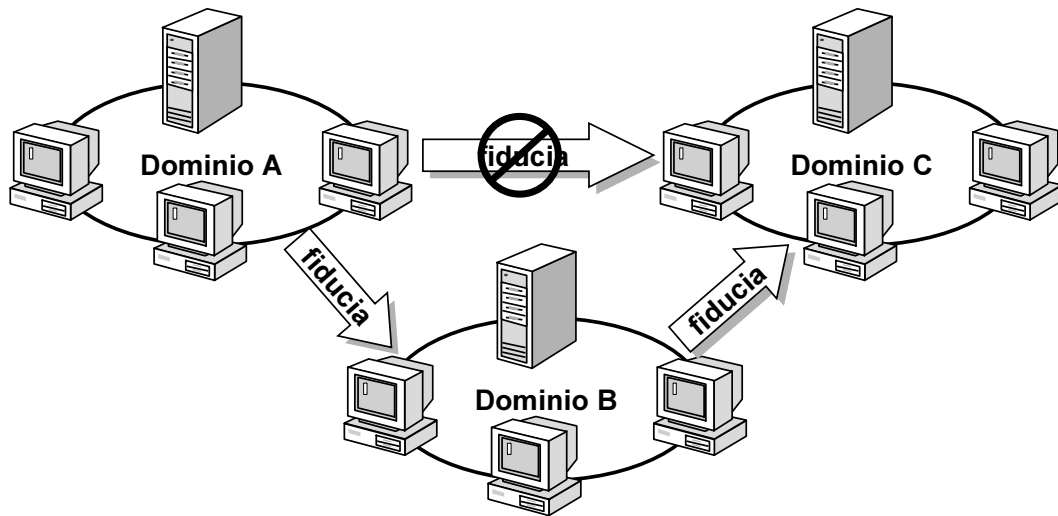


Le relazioni di fiducia sono formulate tra domini: domini trusted e trusting

Relazioni di fiducia

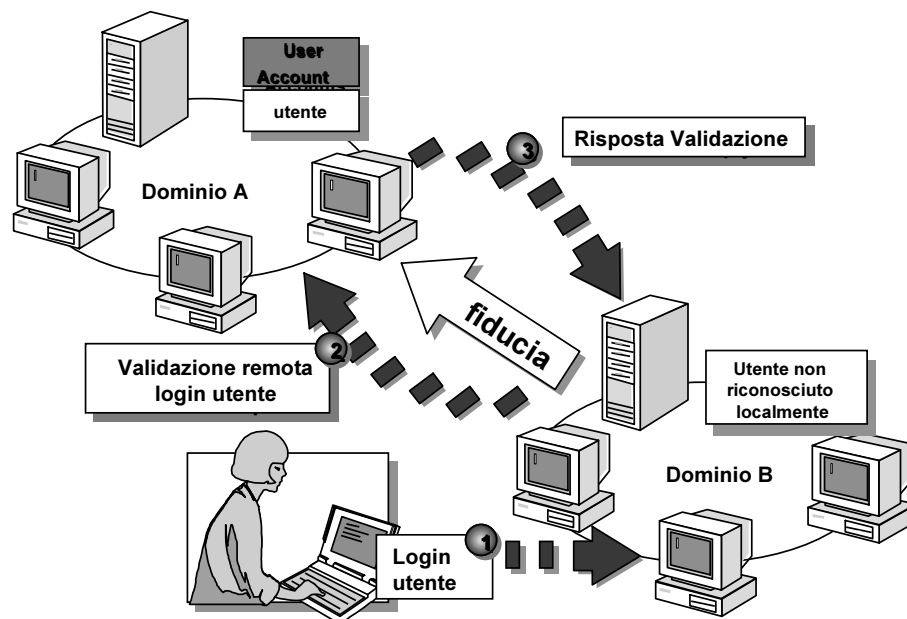


Relazioni di fiducia

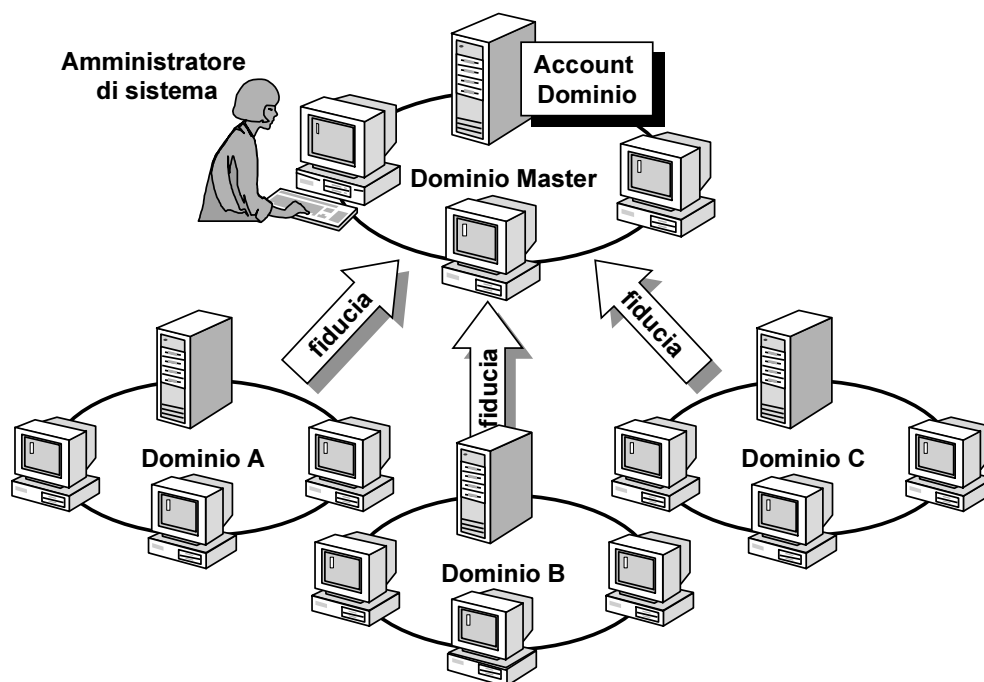


Le relazioni di fiducia non sono transitive

Accesso validato tramite relazione di fiducia

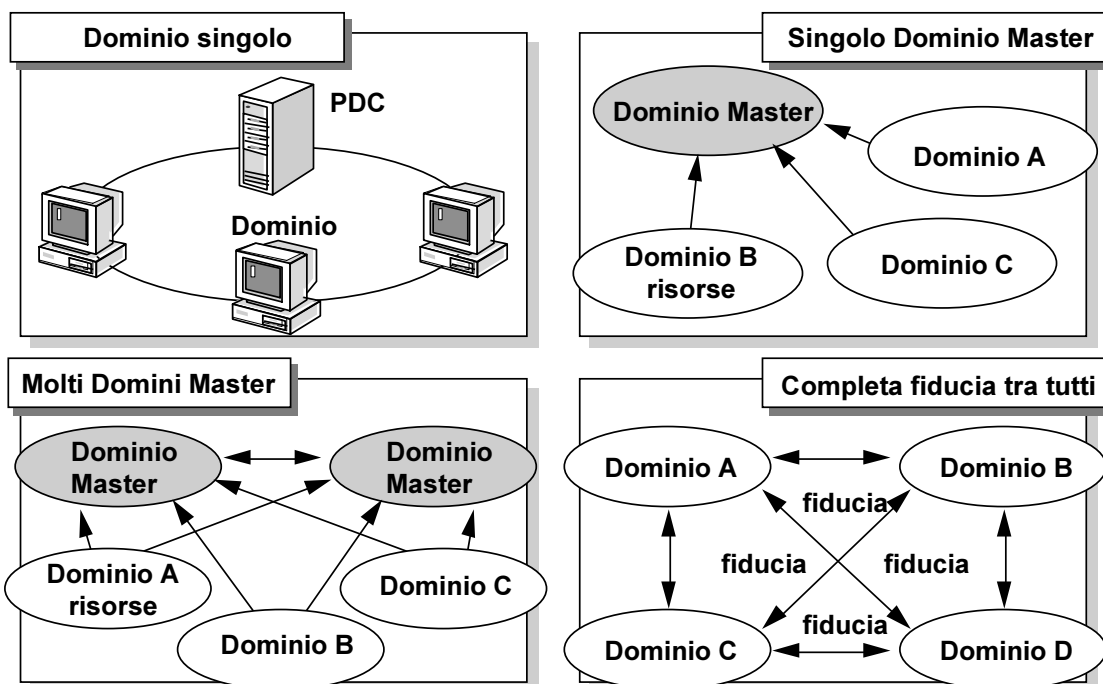


Amministrazione centralizzata



Windows NT -- 75

Modelli di organizzazione



Windows NT -- 76

Windows NT 5 (Windows 2000)

NT 5 estende il sistema Windows verso l'integrazione di sistemi distribuiti e di rete.

Servizi principali previsti in NT 5:

- servizio di **Active Directory**, registra e memorizza le informazioni su tutte le risorse presenti in rete, per facilitare il loro uso, reperimento e gestione da parte degli utenti;
- servizi di **sicurezza** distribuiti, basati su una nuova architettura di sicurezza;
- infrastruttura centralizzata di **amministrazione**.

Windows NT 5

Caratteristiche del servizio **Active Directory**:

- strutturazione **gerarchica** delle informazioni
- **estendibilità** delle strutture dati
- uso di **standard** Internet:
 - Lightweight Directory Access Protocol (LDAP)
 - Domain Name System (DNS)
- **replicazione** delle strutture dati del servizio Active Directory su molti domain controller sincronizzati e aggiornati automaticamente (in sostituzione di Backup Domain Controller)

