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 Microsoft Windows

- **Windows 3.x** è in fase di abbandono ma ha ancora una notevole base di installato, soprattutto nelle grandi aziende (16 bit).
- **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 (ibrido 16/32).
- **Windows NT** è stato pensato come sistema per uso professionale. È un 32bit vero con un'architettura molto robusta che si rivolge allo stesso mercato dei sistemi Unix (comprende un sottosistema conforme a Posix).
- **Windows CE** è destinato al mercato palmari ed embedded (32 bit).
- **Windows 2000** è un'evoluzione di Windows NT (32 bit).
- **Windows XP** aggiunge alcune caratteristiche di 95/98 al di sopra di 2000
- **Windows Advanced Server**, prima versione a **64 bit**, per il processore Intel IA64. Tale versione dovrebbe essere rinominata Windows .NET Server.

NT/2000/XP come principale piattaforma Microsoft per il futuro:

laptop, desktop, server, mono/multi-processori

Windows 95/98 e NT/2000

Differenze NT/2000 (e XP) rispetto a 95/98

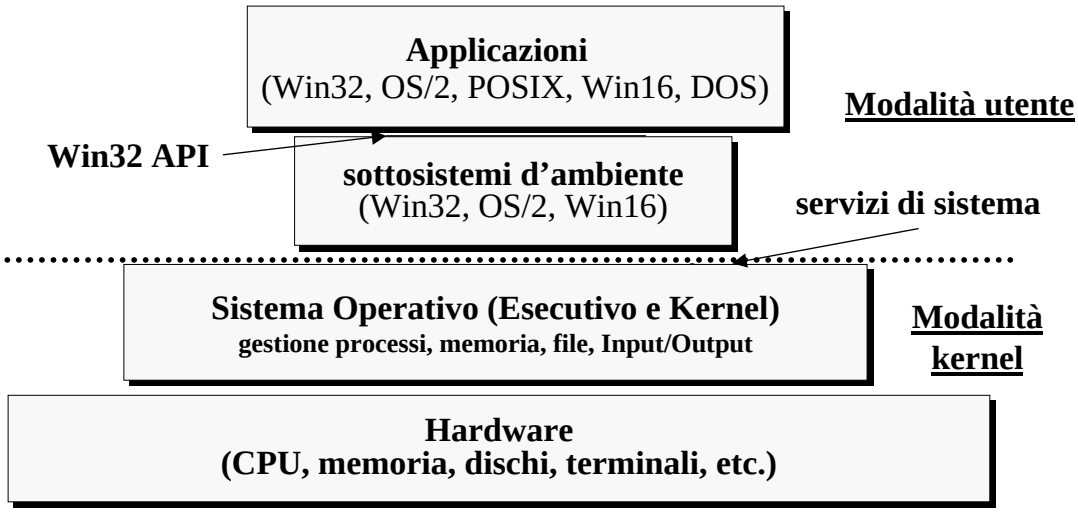
- Supporto ai sistemi multiprocessore (fino a 32 CPU)
- NT portabile, multipiattaforma (NT su DEC alpha, PowerPC, Intel, 2000 solo su Intel) 95/98 solo su Intel x86
- NT/2000 multiutente (autenticazione e controllo accesso ai file)
- NT/2000 è un sistema operativo a 32 bit
- 95 ha molte parti di codice non rientranti (Win16), da eseguire in mutua esclusione (possibile blocco sistema), 2000 ha un kernel totalmente rientrante
- NT/2000 ha protezione address space (diversi processi non possono interferire)
- 95 è 100% compatibile con win3.1 e DOS, NT e 2000 no.

Windows NT (New Technology) e 2000

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)
- Win 2000 estende NT verso l'integrazione di sistemi distribuiti e di rete

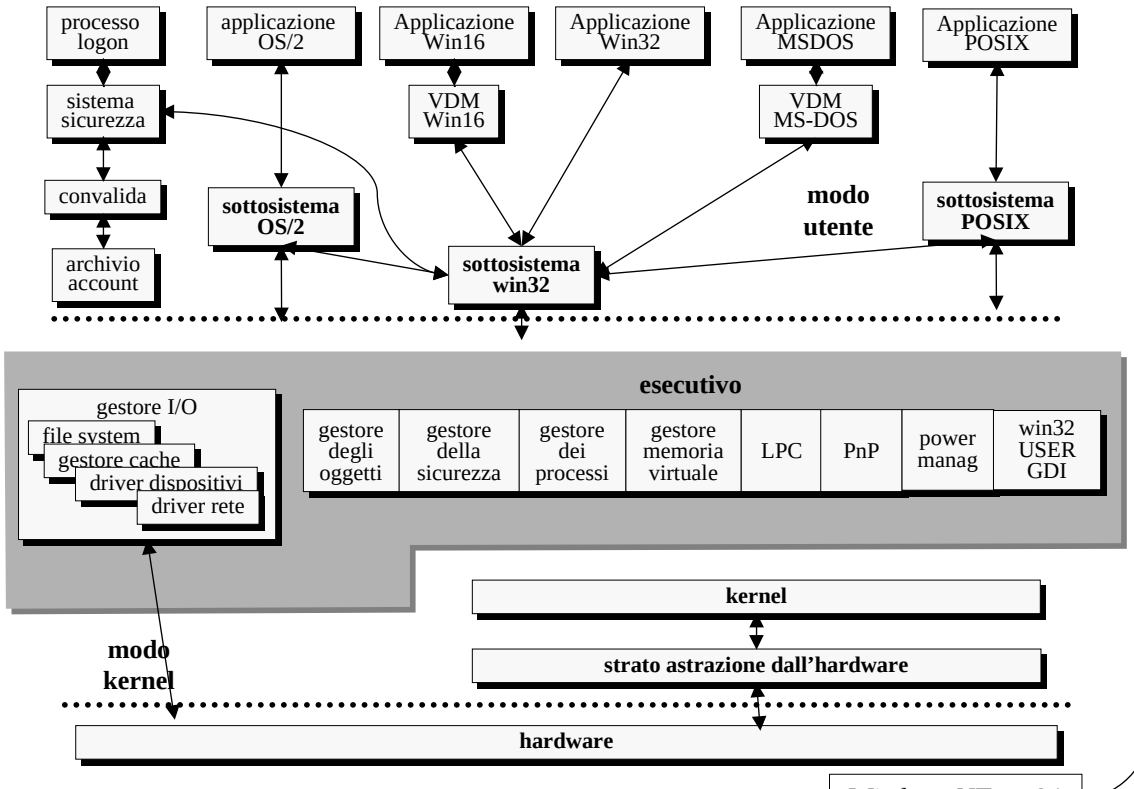
Architettura di Windows NT e 2000



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

Architettura di Windows NT



Principi di progettazione di NT e 2000

Tutte le risorse del sistema sono rappresentate in termini di **oggetti** (oggetto processi, oggetto thread, oggetto file, oggetto memoria) ma non è sistema operativo object oriented

Processi usano oggetti tramite descrittori

Funzioni per la gestione degli oggetti

Progettato per il multiprocessing simmetrico

Molti dei servizi del sistema operativo sono realizzati nello spazio utente (modello Client/Server) ma non è una vera architettura a microkernel (come ad esempio Mach)

Strato di astrazione dell'hardware

L'Hardware Abstraction Layer (HAL):

- nasconde le differenze tra le diverse macchine di una stessa architettura hardware (es. Intel) su cui può eseguire Windows NT (per esempio contiene meccanismi di sincronizzazione per macchine singolo o multi processore)
- realizza una macchina virtuale usata come interfaccia verso l'hardware dagli altri moduli (kernel, esecutivo e anche driver dispositivi)

Kernel

Molti servizi sono realizzati nello spazio utente (modello Client/Server) ma non è vera architettura microkernel (tipo Mach)

Il kernel di Windows NT/2000 fornisce meccanismi e non politiche (eccezione per politiche di scheduling) principalmente per:

- scheduling dei thread
- gestione interrupt, trap ed eccezioni

Il kernel nasconde le differenze architetturali delle varie macchine supportate in NT/2000 (per esempio in gestione interrupt e gestione processi)

Tutto il kernel è fully reentrant, a differenza di grosse parti del kernel (es. GDI) di sistemi Win16. Maggiore stabilità e migliori prestazioni.

A livello di kernel si trovano anche i **device driver** che sono l'interfaccia tra il sottosistema di I/O e l'hardware (scritti principalmente in C)

I processi e i thread

Un **processo** è rappresentato da una struttura dati (executive process block) contenuta nel kernel.

Il process block contiene id utente, id processo, priorità, info per gestione memoria, affinità, quota, etc.

Il processo è dotato di uno spazio di indirizzamento protetto e privato.

Ogni processo può ospitare più **thread** (flussi di esecuzione all'interno di uno stesso spazio di indirizzamento).

Un thread è rappresentato da un descrittore con informazioni: thread ID, priorità, thread affinity, execution time, etc.

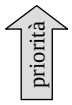
In 2000 esiste il **job object**, un gruppo di processi, su cui si possono imporre azioni: max num di processi, max CPU usage per job o per processo, job scheduling e affinity, default working set…

Lo scheduling in Windows NT/2000

Unità di scheduling in NT/2000 sono i **thread** (non i processi)

NT ha uno **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** (0) (zero-page thread in kernel mode)

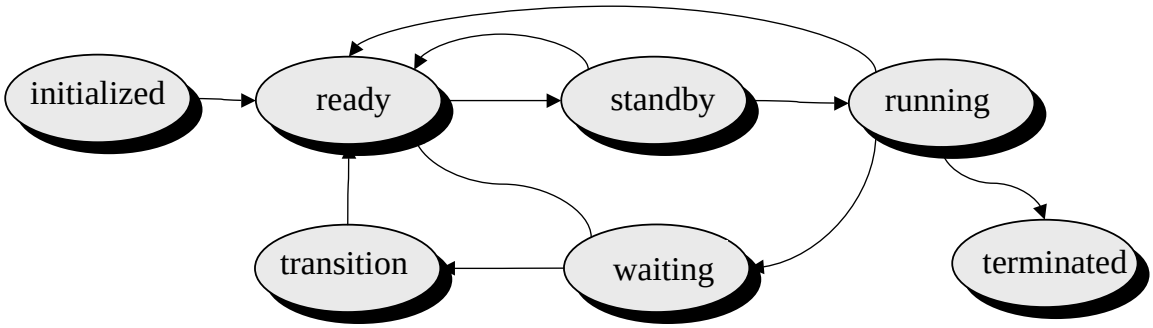


Una coda per ogni livello di priorità con **round robin** in ogni coda.

In ogni istante il kernel esegue il thread (di qualunque classe, di qualunque processo, di qualunque modalità) 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/2000 abbassa la priorità dei thread CPU bound e privilegia i thread interattivi e I/O bound)

Scheduling dei thread

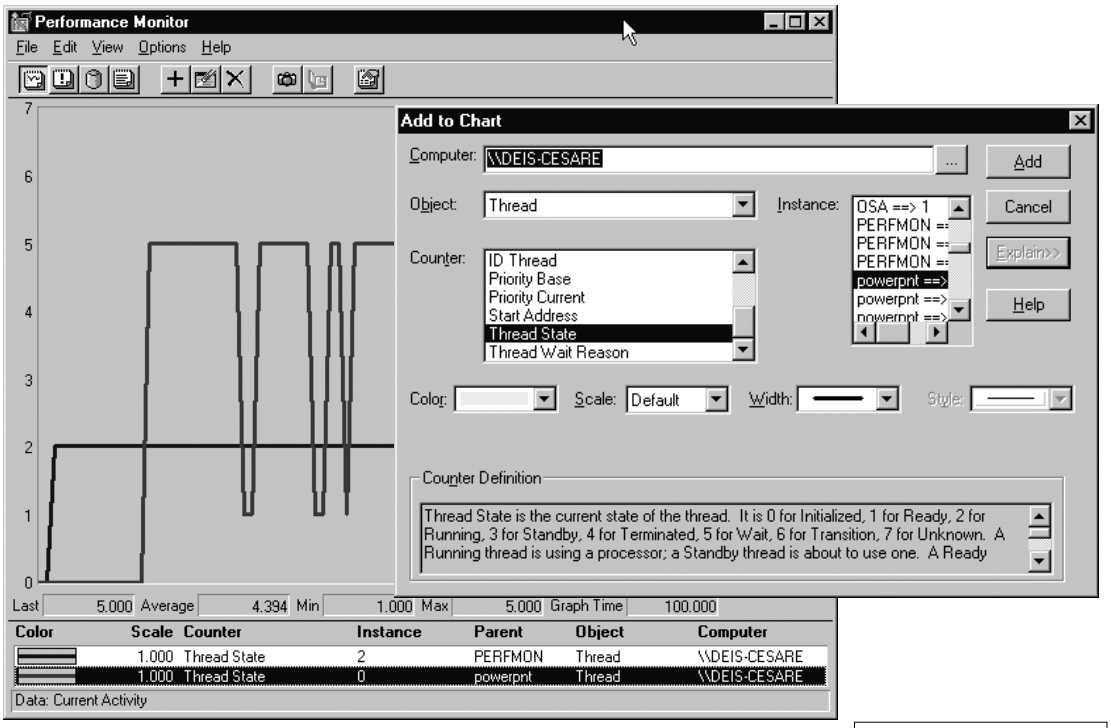


Caso **multiprocessore**:

- 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

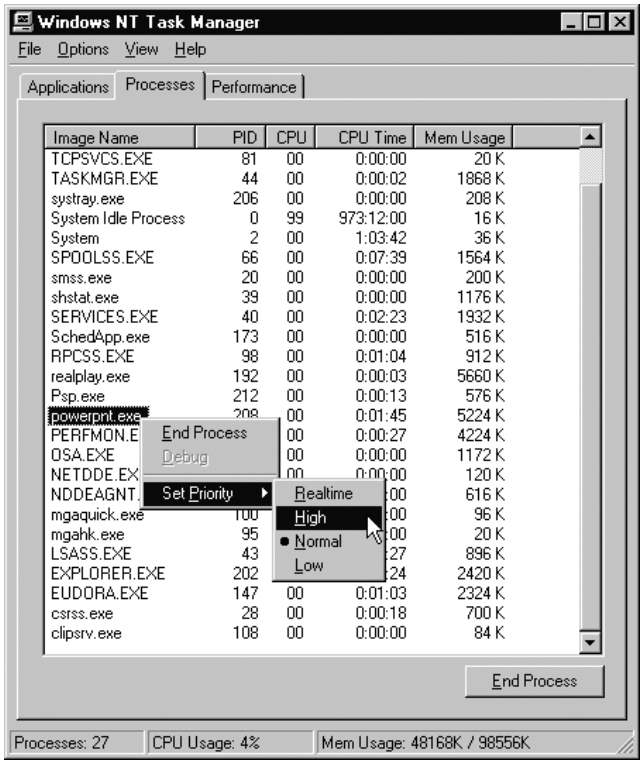
Esempio: visualizzazione stato dei thread

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 (caso NT).



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 (caso NT).



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

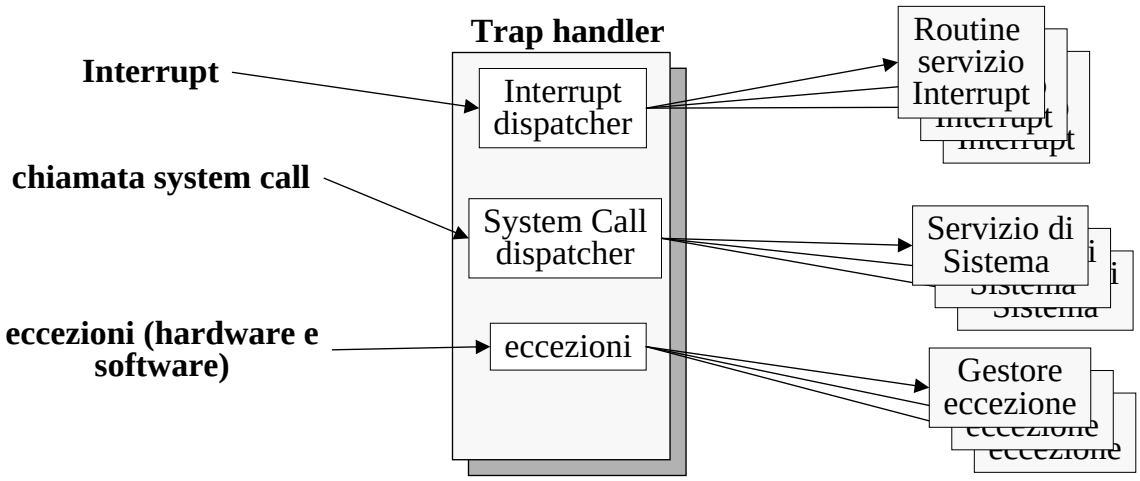
1) interrupt sono eventi asincroni
2) eccezioni sono eventi sincroni
3) system call richiedono servizio al sistema operativo

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

Interrupt ed eccezioni

Kernel disabilita interruzioni nell'esecuzione di sezioni critiche

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

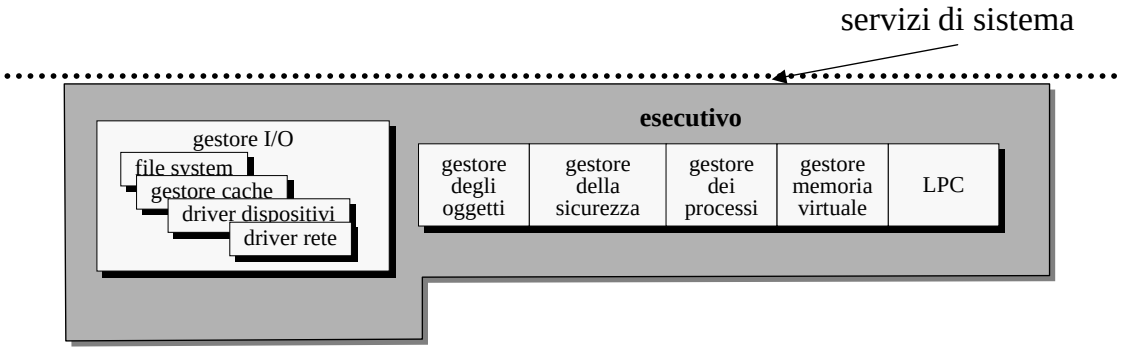
IRQL hanno priorità, interrupt serviti rispettando priorità:

- 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.

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)

La memoria virtuale in NT e 2000

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)

Ovviamente la piattaforma hardware deve avere un **meccanismo di paginazione**

Utilizzo di TLB (Translation Look-aside Buffer) per migliorare prestazioni

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

Per spazi di indirizzamento superiori:

Windows 2000 Advanced Server e Datacenter Server possono sfruttare fino a 8 e 64 GB di memoria fisica, utilizzando le Address Windowing Extensions (sfruttano istruzioni di chipset del Pentium Pro)

Windows Advanced Server è prima versione di Windows a **64 bit**, per il processore Intel IA64 (Itanium). Tale versione dovrebbe essere rinominata Windows .NET Server.

Gestore della memoria virtuale

In un gestore di memoria virtuale si definiscono 3 tipi di politiche:

- politica di **fetch**: quando caricare la pagina da memoria di massa a memoria centrale
- politica di **allocazione**: in quale frame di memoria fisica allocare una pagina di memoria
- politica di **sostituzione**: quando non ci sono frame liberi, quale pagina scaricare dalla memoria per liberare spazio e caricare la pagina che ha generato il page fault (strategie locali e globali)

Windows 2000:

- **fetch**: demand fetching with clustering, caricamento di una pagina e di quelle "vicine". Elevato costo di startup ma migliori prestazioni (località di esecuzione dei programmi)
- **allocazione**: scelta del frame al fine di migliorare prestazioni della cache della CPU
- **sostituzione**: politica locale, working set minimo e massimo per processo. Modifica dinamica del working set per ogni processo a seconda delle esigenze mostrate, cioè del numero di page fault. Su multiprocessori 2000 usa una politica FIFO. Su uniprocessore usa una politica simil LRU (Least Recently Used)

La protezione della memoria

Il gestore della memoria virtuale assicura la protezione della memoria, fondamentale in qualunque sistema multi user e multi tasking.

Assicura che nessun processo possa accedere alla memoria allocata ad altri e che nessun processo in user mode possa accedere a memoria allocata al sistema. In particolare:

- Tutte le strutture dati del kernel sono accessibili solo da processi in kernel mode
- Ogni processo ha uno spazio di indirizzamento separato (a eccezione di eventuali oggetti condivisi esplicitamente)
- Protezione hardware delle pagine marcate read/write, con controllo a tempo di esecuzione
- Anche shared memory object protetti da ACL

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 in Windows 2000

I thread possono essere sincronizzati su molti oggetti, e.g., processi, thread, file, console input, mutex, semafori, eventi, timer. Questi oggetti possono essere in uno stato **signaled** o **unsignaled**:

- un thread si sospende su un oggetto unsignaled (in seguito a esecuzione di *WaitForSingleObject()* su tale oggetto)
- quando l'oggetto entra in stato signaled il kernel sblocca tutti i thread sospesi

Attenzione che:

- mutex è semaforo binario, quando entra in stato signaled sveglia un solo thread
- semafori sono i tradizionali semafori visti in programmazione concorrente
- timer entra in stato signaled dopo uno specificato intervallo di tempo

Sincronizzazione nel kernel:

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

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.

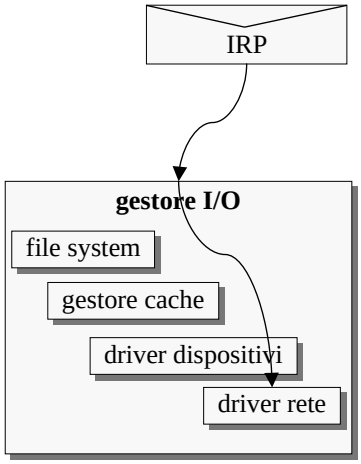
NT ha diverse tecniche di scambio di messaggi, adatte a messaggi di lunghezza diversa

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.



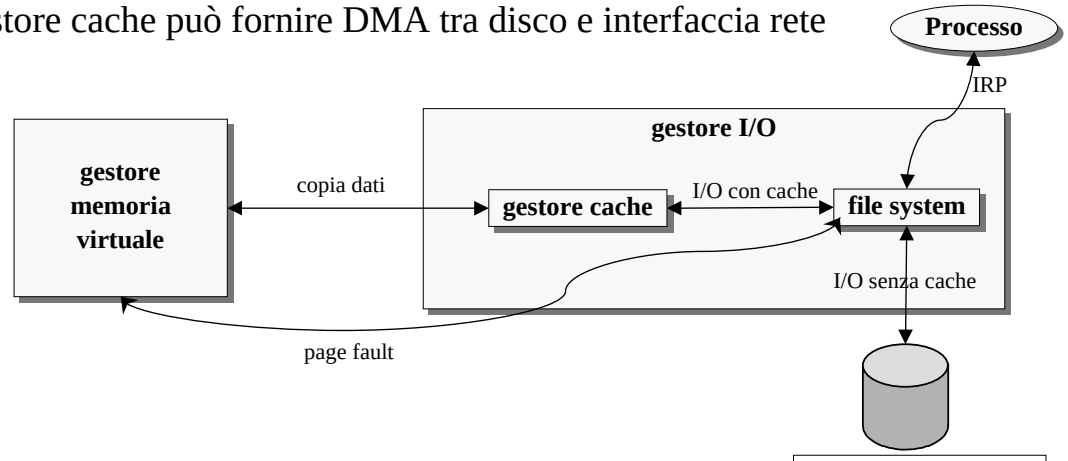
Il gestore della cache

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

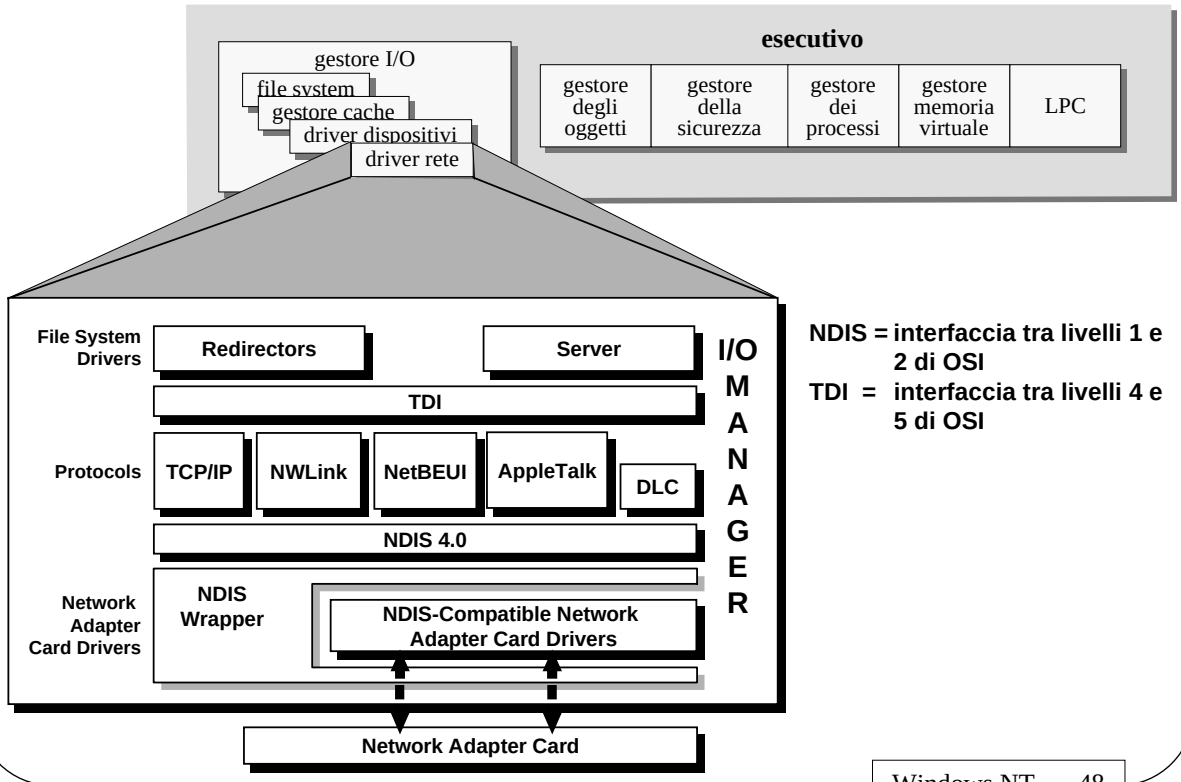
NT riserva per caching fino a 1 GB dello spazio di indirizzamento di 2 GB riservato al SO dal gestore della memoria virtuale. Strategie gestione cache:

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

gestore cache può fornire DMA tra disco e interfaccia rete



Il driver di rete



NDIS = interfaccia tra livelli 1 e 2 di OSI

TDI = interfaccia tra livelli 4 e 5 di OSI

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

La sicurezza in Windows NT

NT 3.51 e NT 4 certificati di classe C2 (livello Orange Book):

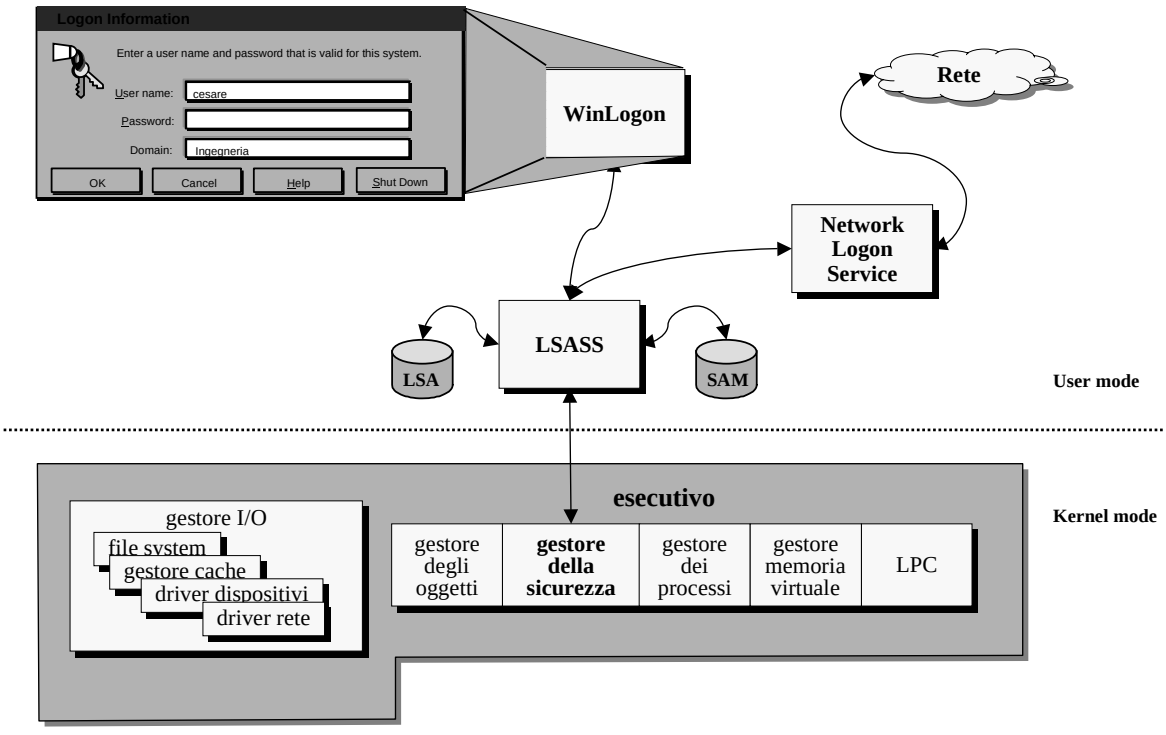
- **secure logon facility**
- **Discretionary Access Control** (DAC), il proprietario di una risorsa definisce una Access Control List che specifica i tipi di accesso da parte di altri utenti e gruppi
- **security auditing**, NT rileva e registra gli eventi ritenuti importanti ai fini della sicurezza
- **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)

Alcune funzioni di classe B:

- **trusted path** con il sistema operativo (ctrl+alt+del)
- **trusted facility management**, supporto per diversi account amministrativi con diversi permessi

La certificazione di un prodotto è un processo lungo e costoso, Windows 2000 è in corso di certificazione secondo il Common Criteria, uno standard di riferimento sviluppato da USA e alcuni paesi europei

Architettura sottosistema di sicurezza di NT/2000



Architettura sottosistema di sicurezza di NT/2000

WinLogon processo a livello utente che si interfaccia con utente a cui chiede username e passwd (inviate a a LSASS), crea tutti i processi richiesti in fase di login

LSASS (Local Security Authority Server) **processo** a livello utente che impone e controlla il rispetto della politica di sicurezza, composto di LSA server e SAM server

LSA (Local Security Authority) verifica accesso utenti

SAM (Security Accounts Manager) contiene le informazioni su utenti, gruppi, password e attributi

Architettura sottosistema di sicurezza di NT/2000

WinLogon viene notificato di una richiesta di login utente in seguito alla pressione dei tasti della SAS (secure attention sequence), tipicamente ctrl-alt-del, per eliminare uso di programmi password-capture (uso di DLL GINA Graphical Identification and Authentication, sostituibile)

WinLogon invia username e password al LSASS e in caso di autenticazione riuscita, crea i processi utente (USERINIT e poi EXPLORER)

LSASS autentica utente e genera un **oggetto d'accesso** (token) con il profilo di sicurezza dell'utente (identificatore di sicurezza di utente, di gruppi e di privilegi, per definire cosa può fare l'utente nel sistema); il token è ereditato da tutti i processi utente successivi

In Win 2000 LSASS può usare modulo di autenticazione microsoft (controllo passwd in SAM) oppure Kerberos 5.

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.): confronta **l'oggetto di accesso** (access token) associato a ogni processo con la **Access Control List** (ACL) dell'oggetto che il processo vuole usare

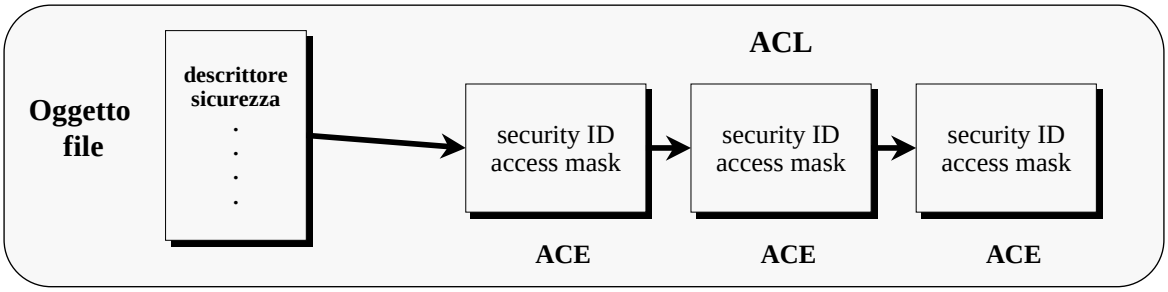
Per motivi di efficienza, autorizzazione eseguita SOLO al momento dell'apertura dell'oggetto (i processi di kernel usano puntatori a oggetti, non handle e quindi by-passano l'autorizzazione)

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



I processi in modalità utente

Vi sono 4 tipi di processi che possono eseguire in **modo utente:**

- Processi speciali di supporto di sistema, come il sottosistema di **accesso e di sicurezza** che autentica un utente tramite password e 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 che l'utente tenta di accedere a un oggetto del sistema.
- **Processi Server** (simili ai demoni Unix) che offrono i servizi NT, come l'event logger
- **Sottosistemi di ambiente**
- **Applicazioni utente** (scritte per win32, win 3.1, DOS, POSIX e OS/2)

Sottosistemi di Ambiente

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

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

Interfaccia di programmazione principale: **Win32**

Altri sottosistemi: **OS/2, POSIX** (solo POSIX.1, no RPC, no socket)

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

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

Gestione di sistemi complessi

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

Problemi:

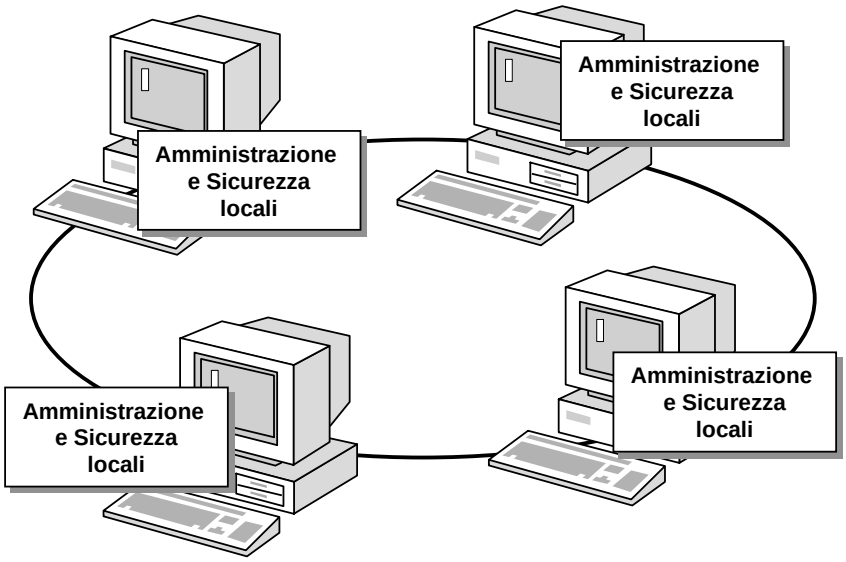
- **eterogeneità** delle piattaforme (varie versioni Windows, Unix, Mac)
- interconnessione di macchine su **scala globale**
- Internet come rete **non sicura**

Ci sono vari **modelli di organizzazione** di una rete di calcolatori, per offrire servizi multipiattaforma e operazioni sicure.

Nel mondo Windows si parla di:

- Modello a Workgroup
- Modello a Dominio (NT)
- Modello Active Directory (2000)

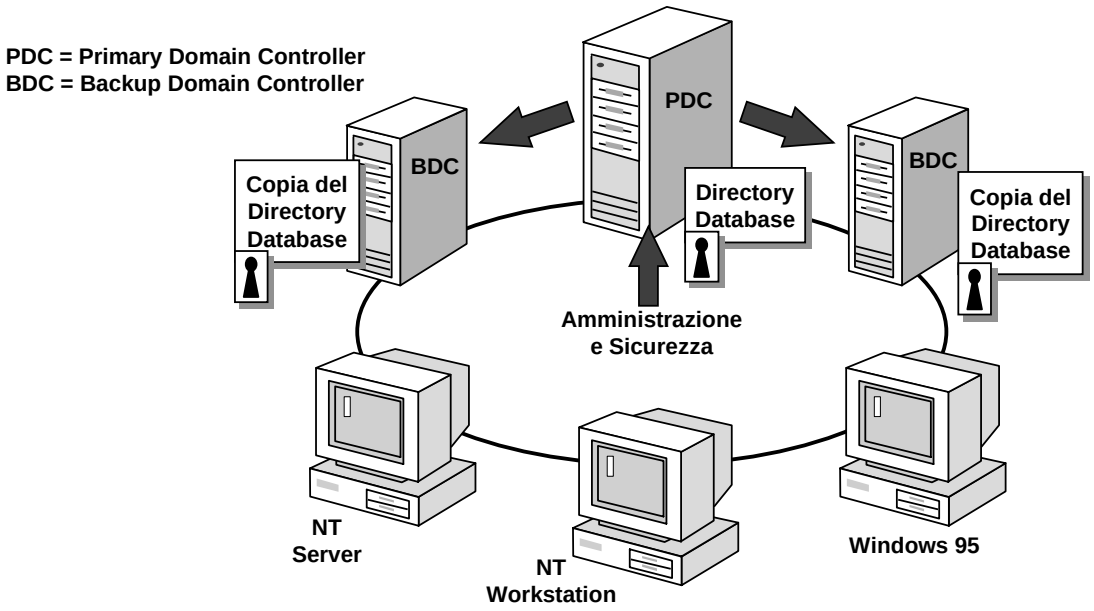
Il Modello Workgroup



Ogni macchina è gestita autonomamente:

- difficile imposizione politiche globali
- alti costi di gestione (no scalabilità)

Il Modello a Dominio (NT)

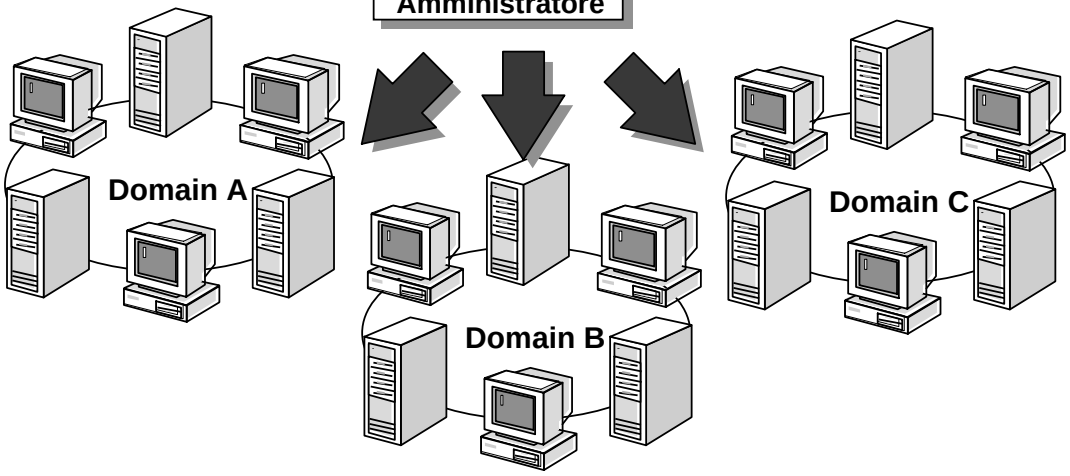


Un dominio è un gruppo di stazioni di lavoro e di Server NT che condividono una politica di sicurezza e il database degli utenti (in pratica si centralizza la gestione dell'autenticazione degli utenti)

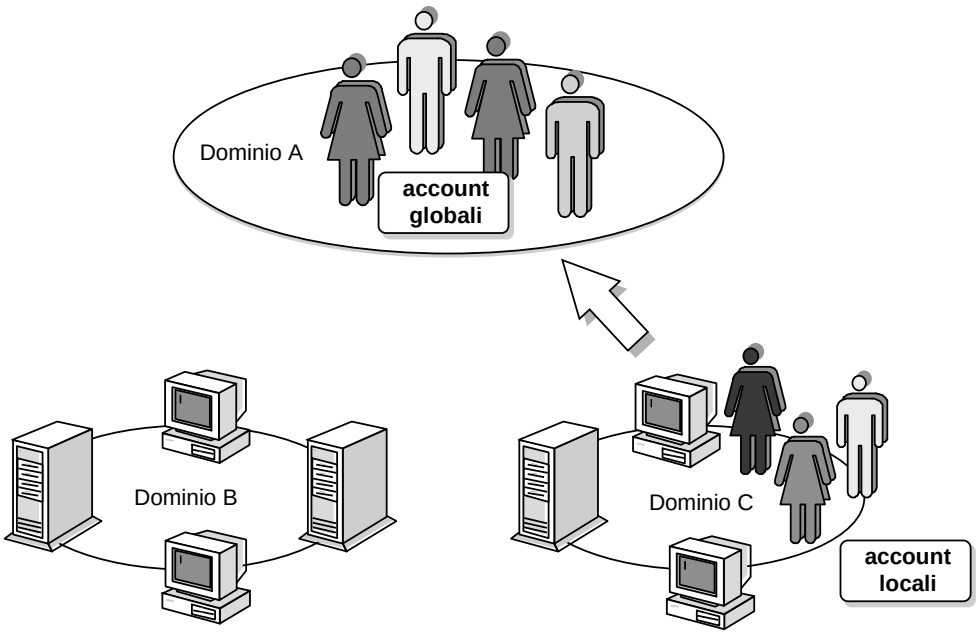
Gestione di reti complesse

Obiettivi:
Unico username
Accesso a tutte le risorse

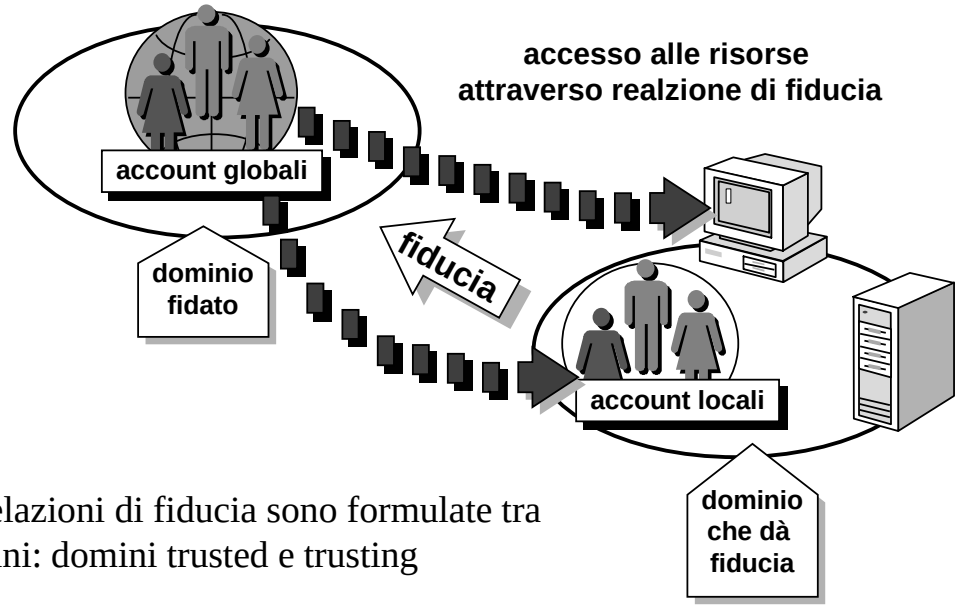
Amministratore di rete (caso centralizzato)



Tipi di account gestiti in NT

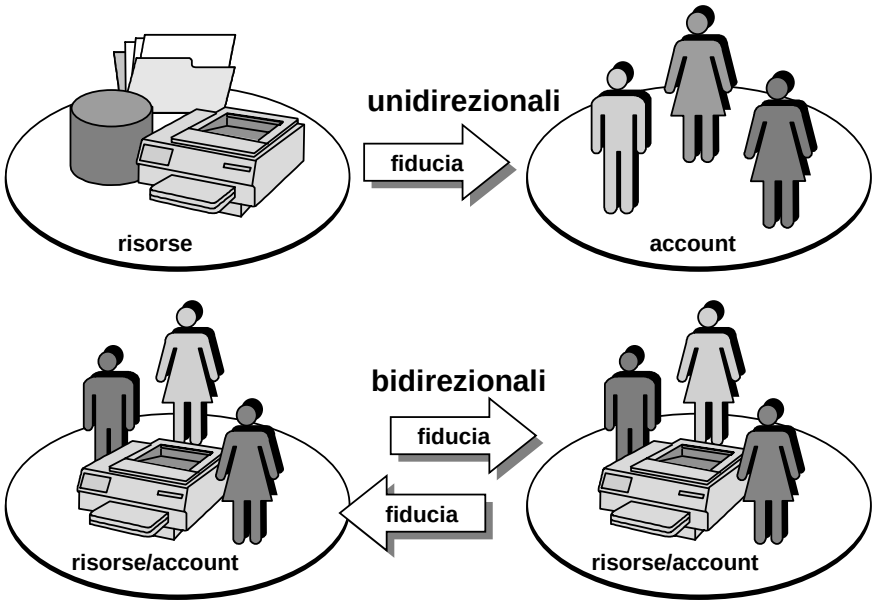


Relazioni di fiducia

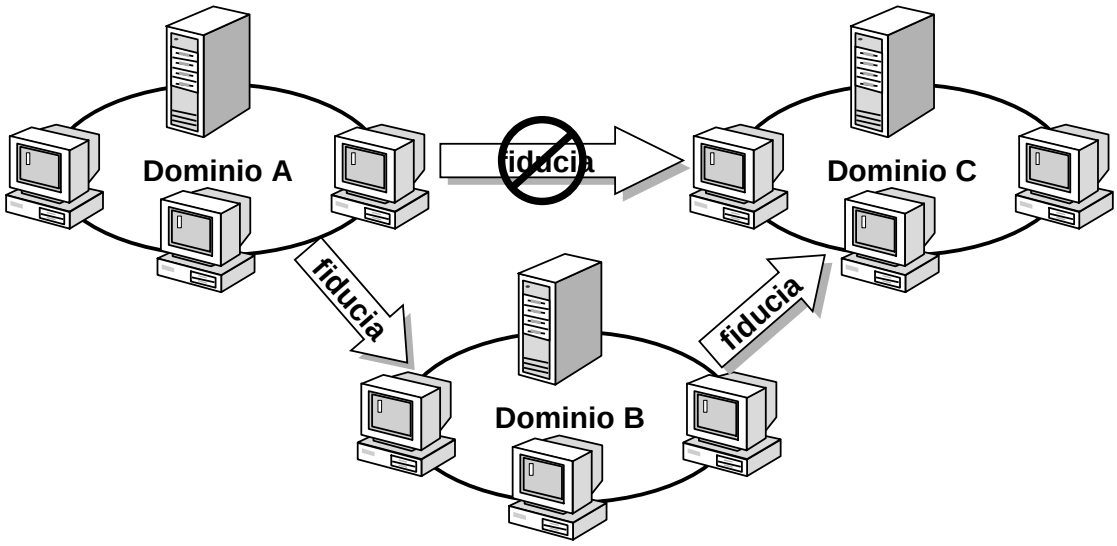


Le relazioni di fiducia sono formulate tra domini: domini trusted e trusting

Relazioni di fiducia

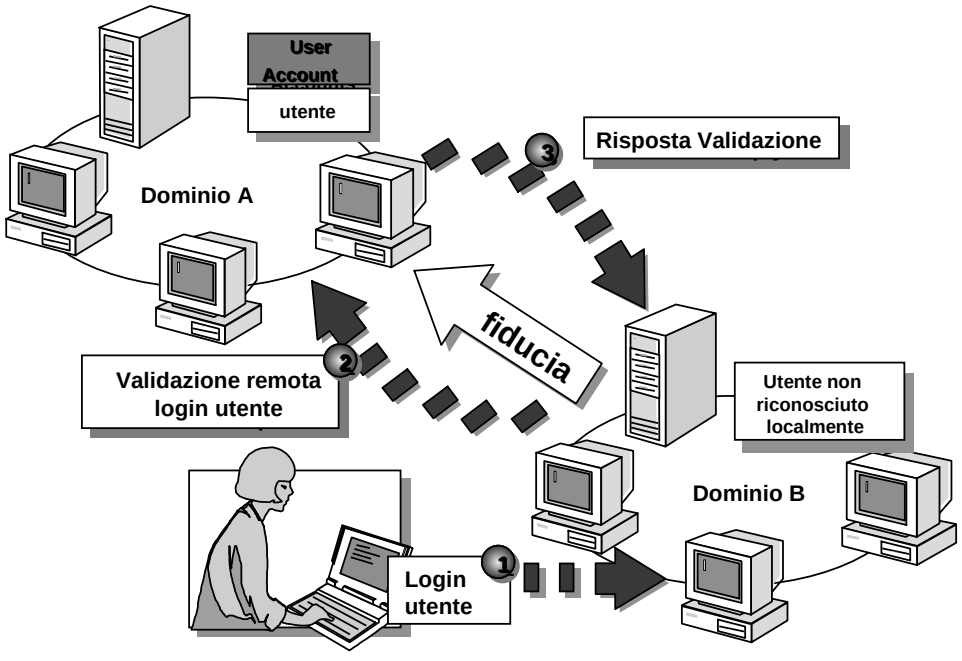


Relazioni di fiducia

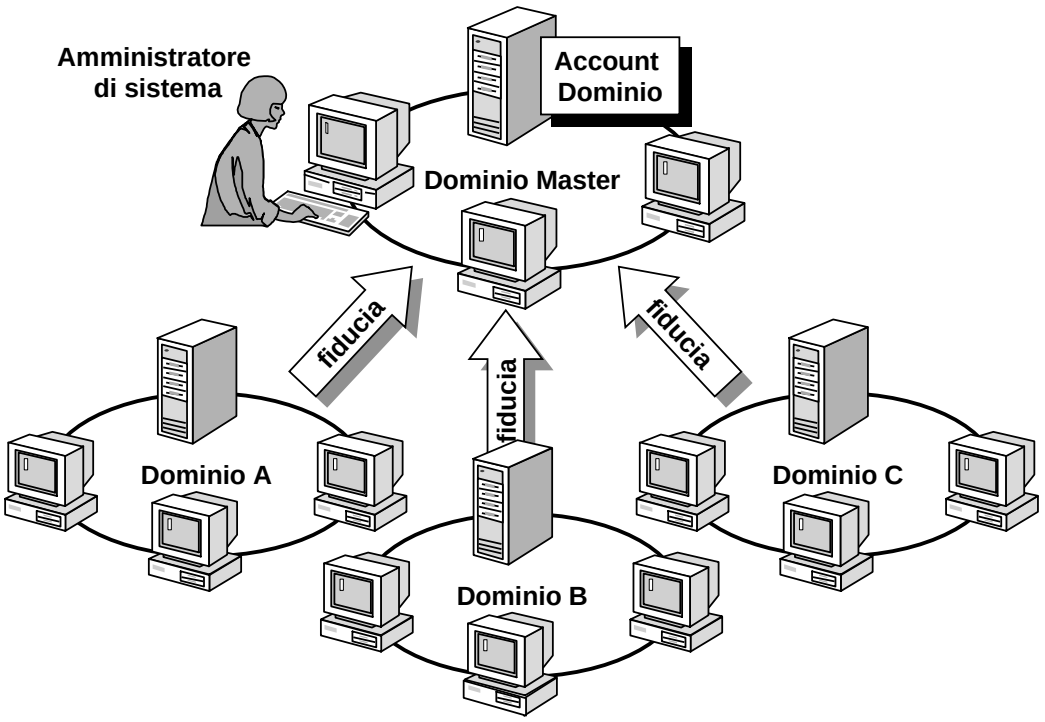


Le relazioni di fiducia non sono transitive

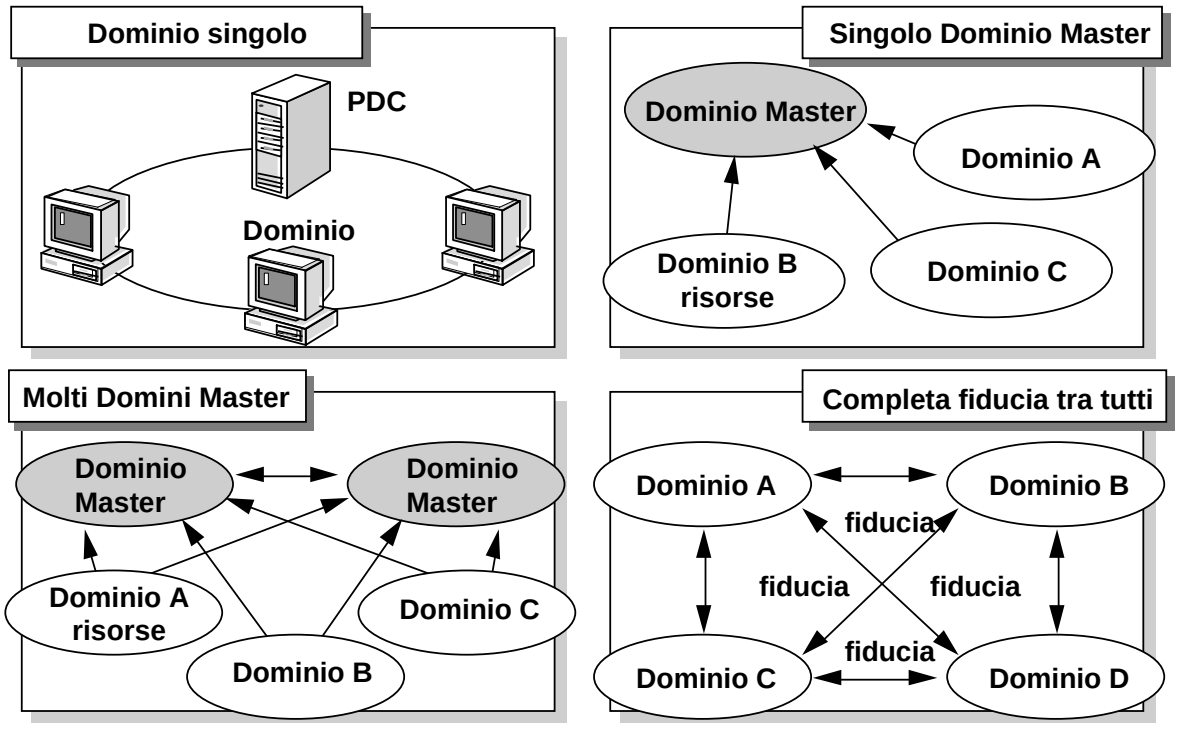
Accesso validato tramite relazione di fiducia



Amministrazione centralizzata



Modelli di organizzazione



Windows 2000 Active Directory

La principale differenza tra NT 4 e 2000 è l'introduzione del servizio di **Active Directory**.

Active Directory contiene tutte le informazioni su tutte le risorse del sistema distribuito. Questo permette una gestione centralizzata, molto più efficiente, facile e sicura.

Active Directory interrogato per eseguire autenticazione, per ottenere informazioni di configurazione dispositivi (es. stampanti), informazioni su utenti (es. telefono, e-mail...), etc.

Active Directory è un servizio di directory, cioè una specie di database, che però è pensato, progettato e realizzato per essere fruito via rete e ottimizzato per le operazioni di lettura.

Windows 2000 Active Directory

Esempio di risorse memorizzate in un Active Directory server:

- username e password di tutti gli utenti
- la struttura e l'organizzazione (gerarchica) del dominio
- le informazioni per la configurazione delle risorse (stampanti, file system, etc.)
- politiche di sicurezza

Windows 2000 Active Directory

Le informazioni in Active Directory sono organizzate **gerarchicamente**, in domini e sottodomini (organizzazione simile a DNS).

L'organizzazione è facilmente estendibile (facile aggiunta di oggetti).

Active Directory fa uso di **standard Internet**, quali il Lightweigth Directory Access Protocol (LDAP) e il Domain Name System (DNS).

Active Directory fornisce una gestione delle relazioni di **fiducia** inter-domini molto più semplice di NT (relazioni biunivoche e transitive).

Le strutture dati nel server Active Directory possono essere replicate su molti **domain controller** sincronizzati e aggiornati automaticamente (multimaster replication).