

Laboratorio di Sistemi Operativi

LAB1

Installazione Windows e relativi
ambienti di lavoro

Note

- casi di studio: Unix/Linux, Windows 2000
- installazione S.O. e ambienti di lavoro
- programmazione (system call, API) in C
- modalità di esame:
 - esercizi programmazione nello scritto
 - installazione S.O. in silab
 - realizzazione di uno degli esercizi dello scritto

Note (2)

- documentazione:
 - Testo Tanenbaum
 - Lucidi sul sito
 - man per syscall Unix
 - <http://msdn.microsoft.com> per api win32

Agenda

- Panoramica e storia di Windows
- Installazione Windows
- Installazione strumenti di sviluppo
 - Visual Studio
 - Tools di monitoraggio e debugging
- Programma d'esempio

Una citazione...

“Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.”

Bill Gates 1976

We are still waiting ...

Piccola cronologia

- 1981 MSDOS 1.0 - 16 bit, command line, single user, Intel 8088
- 1985 Windows 1.0 - GUI su MSDOS; tutti i proc nello stesso address space
- 1993 Windows NT 3.1 (progetto iniziato nel 1988)
 - 32 bit (puro), multiplatforma (Intel, Alpha, MIPS), OS/2, POSIX
- 1995 Windows 95: trasferimento da MS-DOS a Win
 - 32 bit (quasi), virtual memory senza protezioni, multiprogramming, process management
 - Win98: kernel non rientrante: 1! mutex per protezione
- 1996 Windows NT 4.0
 - progettato da Cutler (VMS); GUI compliant con Win95; aggiunta aspetti security e affidabilit ; scritto quasi interamente in C
- 2000 Windows 2000 (derivato da NT)
 - vero 32-bit; protezioni; multi-threading; demand paged; plug & play; non ha MS-DOS; rilasciato in 4 versioni
- 2001 Windows XP

Windows NT

Item	Windows 95/98	Windows NT
Full 32-bit system?	No	Yes
Security?	No	Yes
Protected file mappings?	No	Yes
Private addr space for each MS-DOS prog?	No	Yes
Unicode?	No	Yes
Runs on	Intel 80x86	80x86, Alpha, MIPS, ...
Multiprocessor support?	No	Yes
Re-entrant code inside OS?	No	Yes
Plug and play?	Yes	No
Power management?	Yes	No
FAT-32 file system?	Yes	Optional
NTFS file system	No	Yes
Win32 API?	Yes	Yes
Run all old MS-DOS programs?	Yes	No
Some critical OS data writable by user?	Yes	No

Windows 2000 (2)

Year	AT&T	BSD	MINIX	Linux	Solaris	Win NT
1976	V6 9K					
1979	V7 21K					
1980		4.1 38K				
1982	Sys III 58K					
1984		4.2 98K				
1986		4.3 179K				
1987	SVR3 92K		1.0 13K			
1989	SVR4 280K					
1991				0.01 10K		
1993		Free 1.0 235K			5.3 850K	3.1 6M
1994		4.4 Lite 743K		1.0 165K		3.5 10M
1996				2.0 470K		4.0 16M
1997			2.0 62K		5.6 1.4M	
1999				2.2 1M		
2000		Free 4.0 1.4M			5.8 2.0M	2000 29M

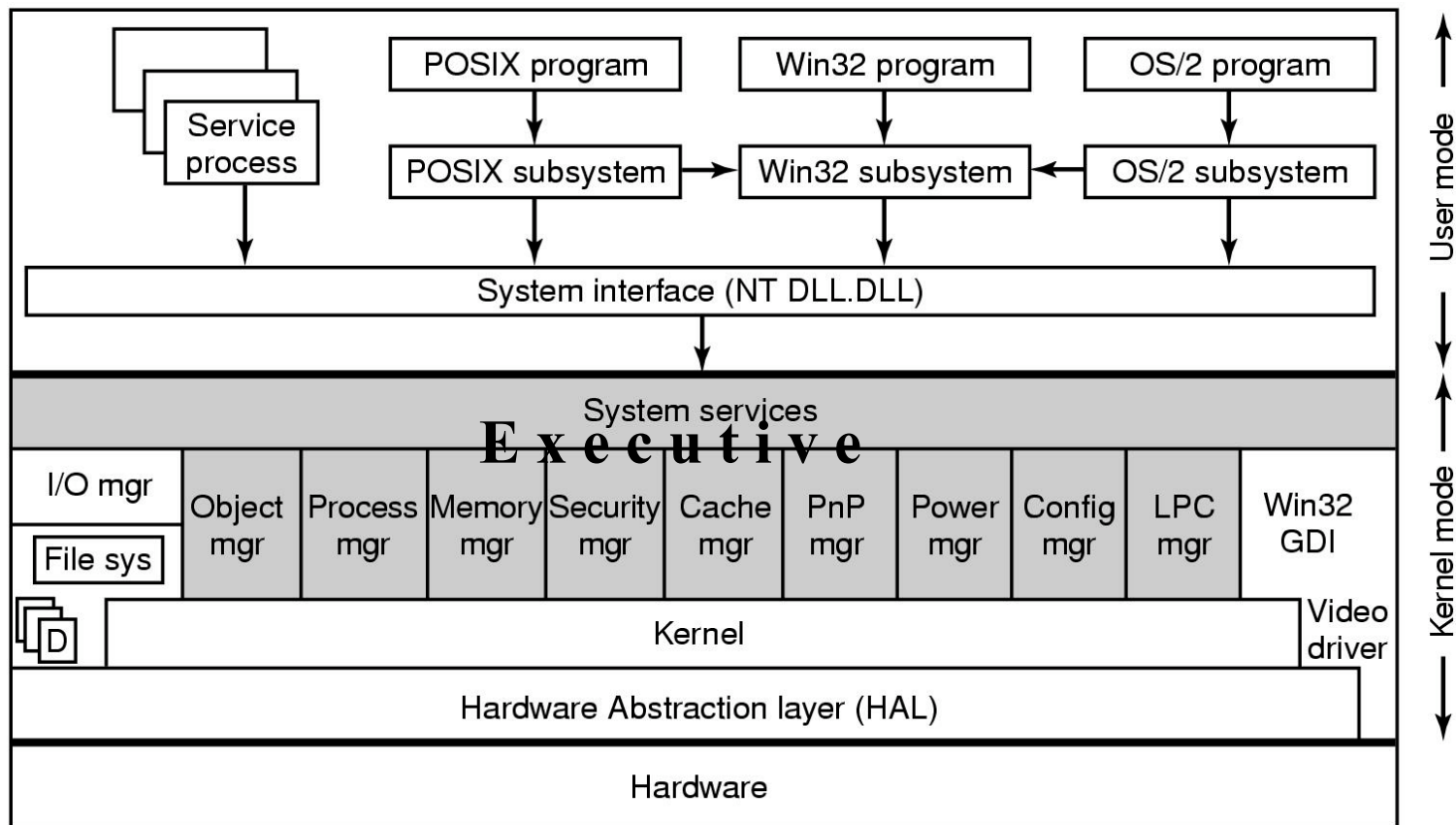
confronto in #linee codice

Win: politica di sviluppo/progettazione naif...

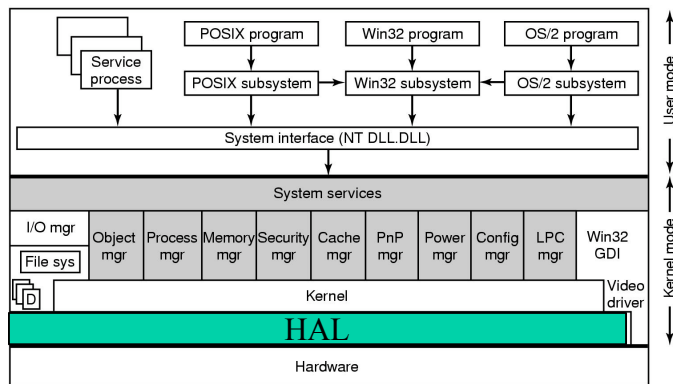
programmazione Win2K

- win32 API: interfacce a syscall
 - syscall mai rese pubbliche
 - API per gestione memoria e file, security, ifc grafica
 - numerose API per lo stesso scopo, con parametri differenti
 - API +/- standard da Win95
 - ma possibili comportamenti differenti o API non supportate da tutti sistemi
- ogni chiamata torna handle a “oggetto”
 - no inheritance e polimorfismo

The Operating System Structure

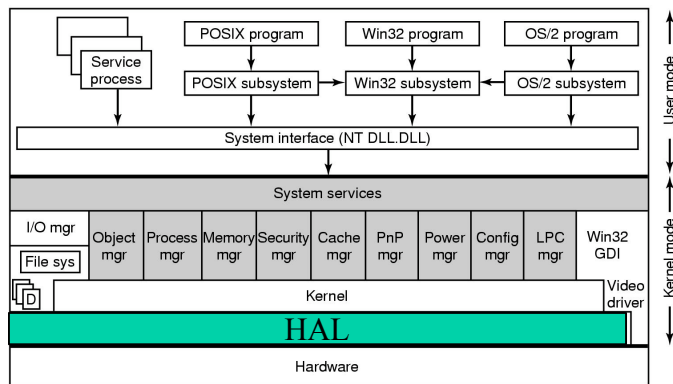


client-server model



Hardware abstraction layer

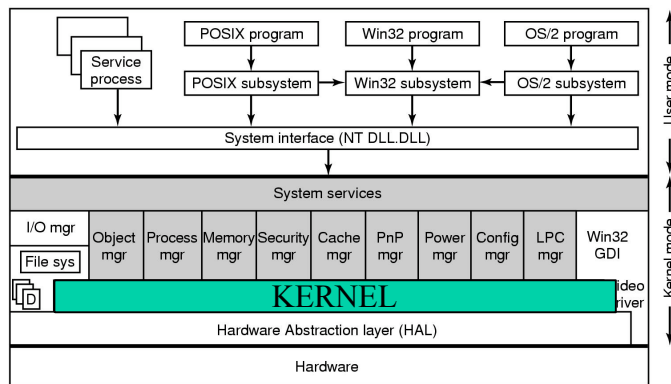
- Nata dall'esigenza di supportare diverse piattaforme (Intel x86, DEC Alpha, MIPS, PowerPC,...)
- driver in C e assembly
- Hal.dll
 - Ce ne sono diversi sul CD di installazione, ma solo uno viene copiato sul disco di sistema
 - Halmps.dll Multiprocessor
 - Halapci.dll Advanced Configuration and Power Interface
 - Halsp.dll Compaq System Pro



Hardware abstraction layer

- Funzioni lettura/scrittura device register:
 - Forniscono astrazioni circa I/O mappato in memoria contro I/O ports
- Indirizzi system-wide per periferiche e interrupt
 - Forniscono un identificativo unico, esente da conflitti sui diversi bus
 - gestione uniforme interrupt
- Gestione DMA, clock, timers, BIOS, CMOS

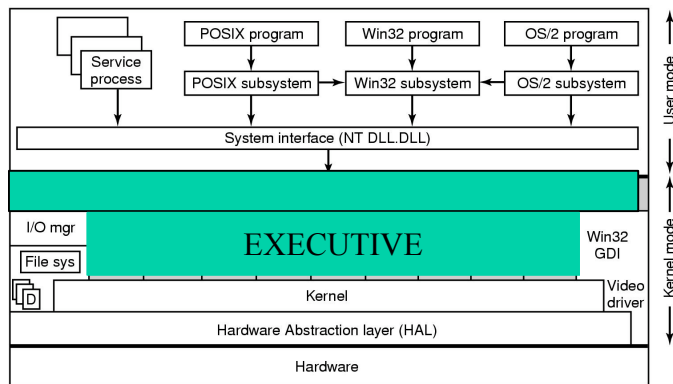
DirectX bypassa HAL (prestazioni)



Kernel

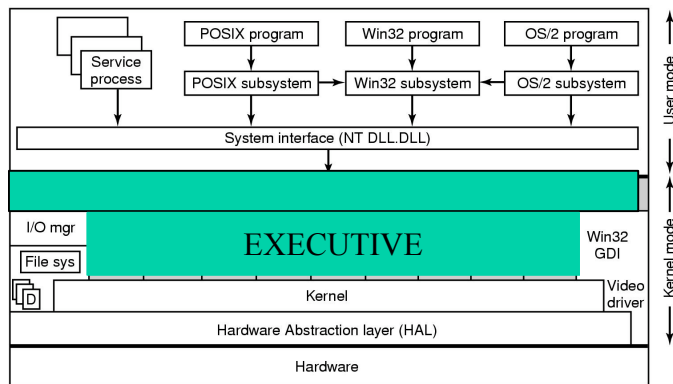
(dizione Microsoft)

- Astrazione dell'architettura hardware ad un livello superiore rispetto ad HAL
- Ntoskrnl.exe
 - Meccanismo di context switching e il codice per il thread scheduling
 - Kernel Objects
 - Supporto (creazione, per oggetti usati dall'Executive per costruire gli oggetti *visibili* in user mode
 - **Control** objects: interrupt, processi, Deferred Procedure Call
 - DPC ritarda exec parte non critica di interrupt routine
 - **Dispatcher** objects: thread, mutex, events, waitable timers



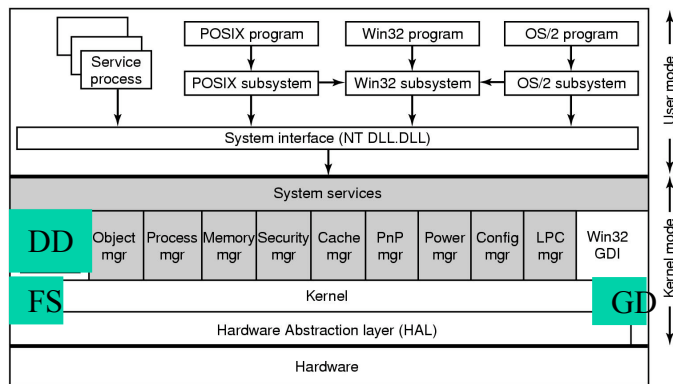
Executive

- in C, indipendente dall'architettura
- System Services: interfaccia
 - Funzioni chiamabili da user mode esportate attraverso **Ntdll.dll** (non tutte documentate)
- Components:
 - Funzioni chiamabili da kernel mode (uso interno all'executive)
 - Gestiscono le politiche di gestione di diverse risorse del sistema



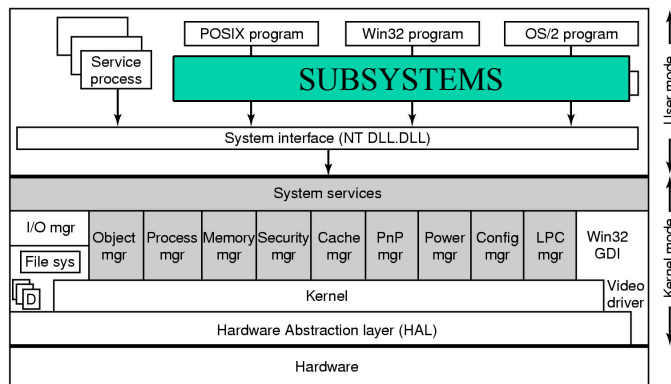
Executive

- object mngr: tiene traccia di tutti gli oggetti
- I/O mngr: gestisce I/O, fornisce servizi generici indipendenti da periferiche
- process mngr: gestisce processi e thread
- memory mngr: per demand paging e protezioni
- security mngr: mecc. per protezioni liv. C2
- cache mngr: caching per multipli f.s.
- plug&play mngr (thread): riceve notifica di nuove periferiche e carica driver opportuno
- power mngr (thread): controlla consumo batterie
- configuration mngr: gestisce il *registry*
- local procedure call mngr: per IPC locale



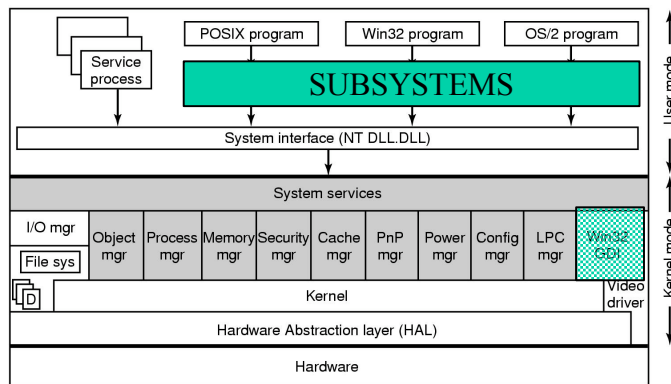
Device Driver

- non sono interni al ntoskrnl.exe : caricati dinamicamente
- sono di vari tipi:
 - Bus Driver
 - gestiscono device che possono avere device figli (PCI, PCMCIA, USB)
 - Function Driver
 - Hardware device driver
 - controllano particolari device tramite HAL
 - File system drivers
 - Filter Driver
 - File System **filter** device driver
 - per attuare disk mirroring, encryption intercettando le chiamate I/O



Environment Subsystems

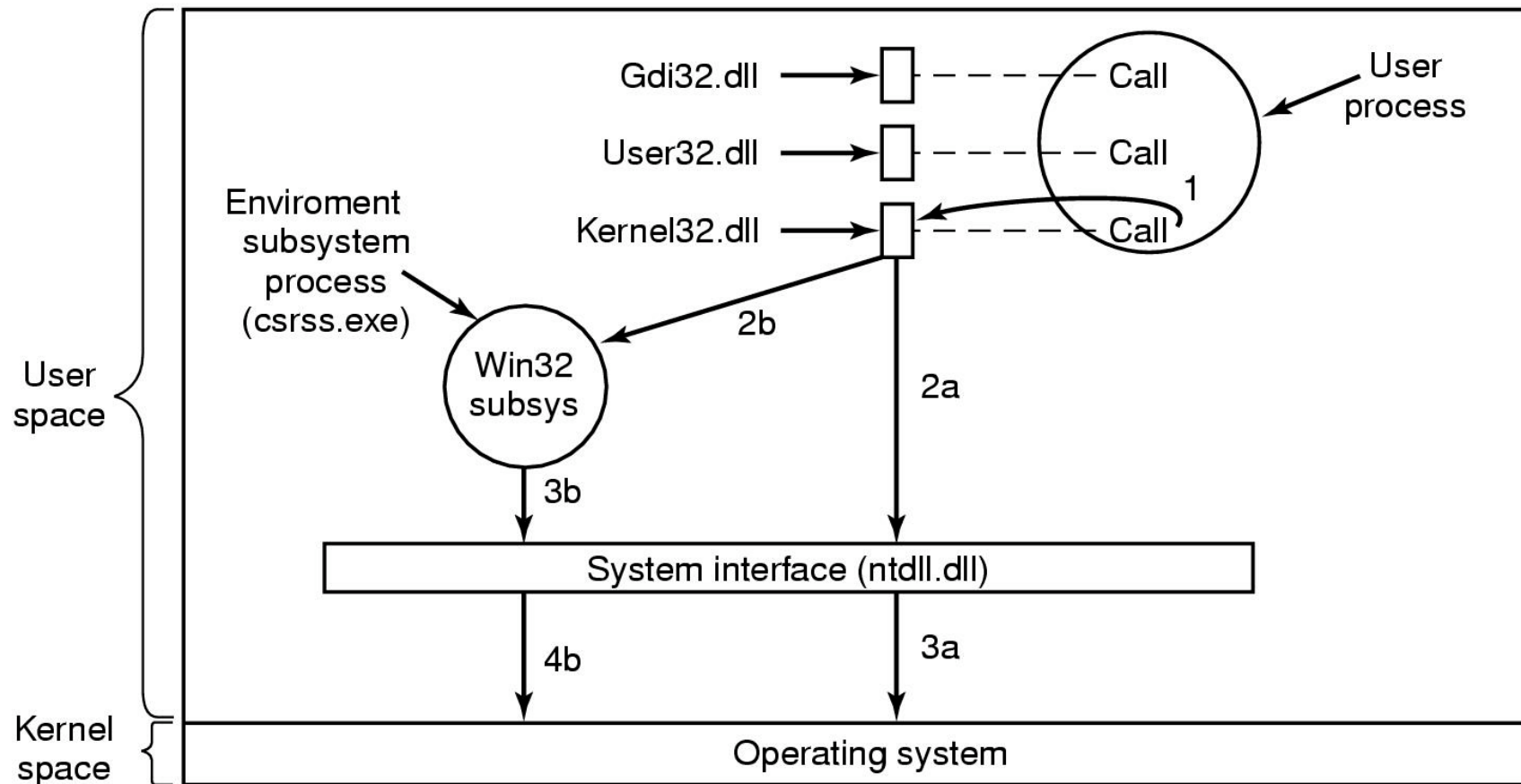
- Corrispondono alle interfacce (API) ufficiali (documentate) implementate ed usabili in WinNT
- NT nasce come successore di OS2
- deve supportare POSIX perche' richiesto dal governo americano
- ma... l'unico che funziona e' Win32



Win32 Environment Subsystem

- E' indispensabile per il funzionamento di Windows
- supportate 3 classi API: Win32, Posix, OS/2
- Csrss.exe
 - console text windows
 - parte della Virtual DOS Machine
 - qualche altra funzione: getTempFile
- Win32k.sys
 - ora in kernel mode
 - GDI (Graphic Device Interface)
 - Window manager
- subsystem DLLs (Kernel32.dll, User32.dll, Gdi32.dll)
 - rimappano le win32 API su kernel system services (Ntoskrnl.exe. Win32k.sys), per condivisione codice tra procs

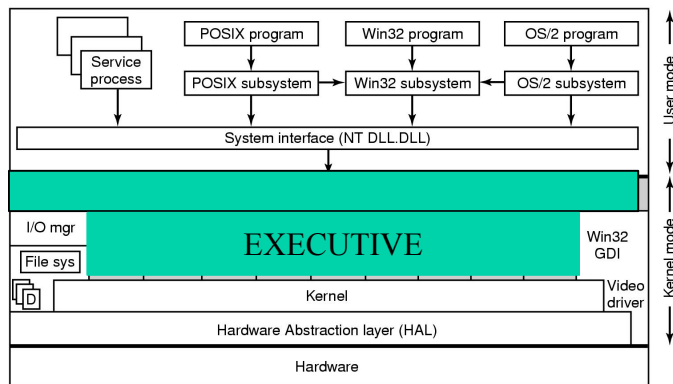
Win32 Environment Subsystem (1)



Vari modi di implementare funzioni Win32 API

Funzioni esportate

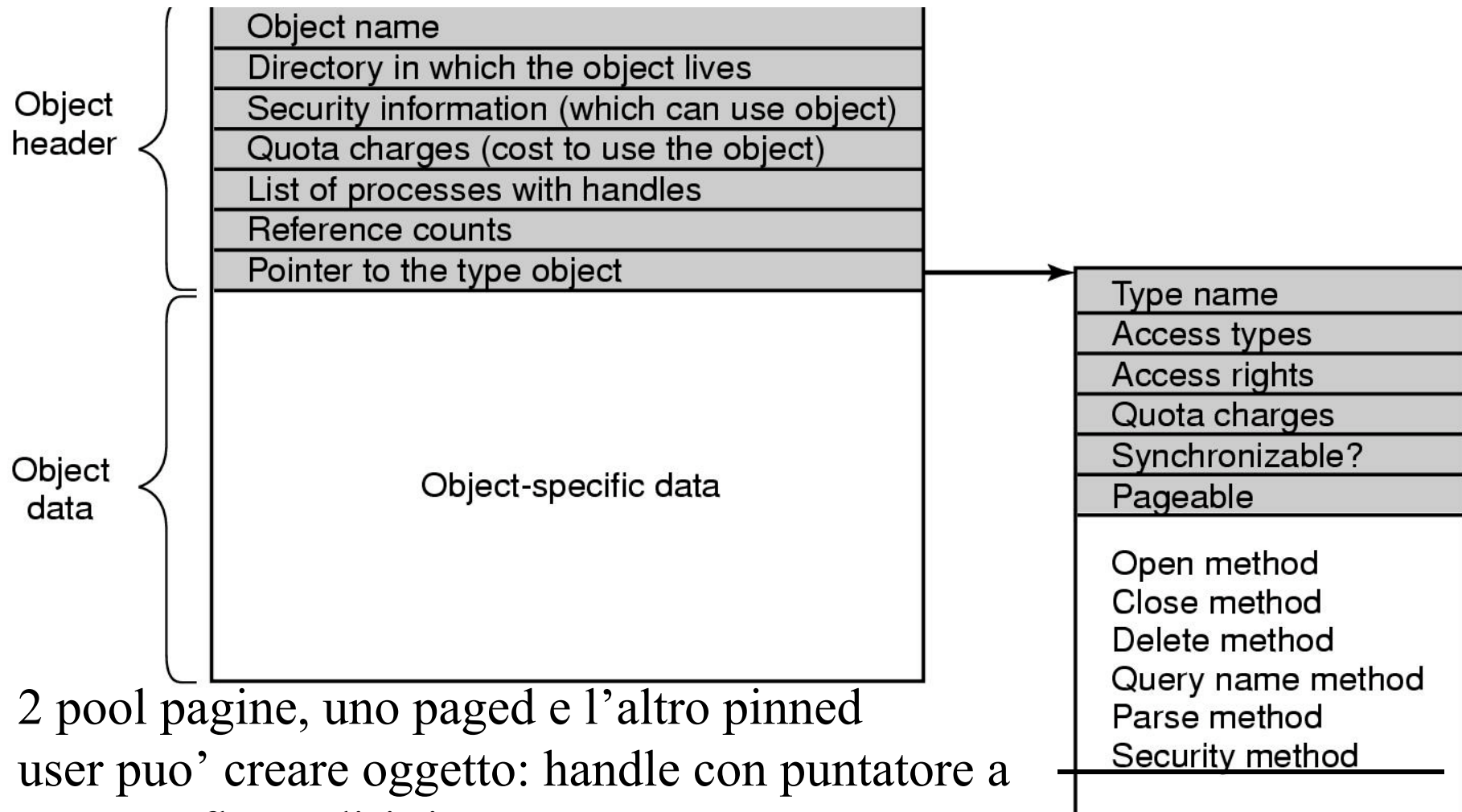
File	Mode	Fcns	Contents
hal.dll	Kernel	95	Low-level hardware management, e.g., port I/O
ntoskrnl.exe	Kernel	1209	Windows 2000 operating system (kernel + executive)
win32k.sys	Kernel	-	Many system calls including most of the graphics
ntdll.dll	User	1179	Dispatcher from user mode to kernel mode
csrss.exe	User	0	Win32 environment subsystem process
kernel32.dll	User	823	Most of the core (nongraphics) system calls
gdi32.dll	User	543	Font, text, color, brush, pen, bitmap, palette, drawing, etc. calls
user32.dll	User	695	Window, icon, menu, cursor, dialog, clipboard, etc. calls
advapi32.dll	User	557	Security, cryptography, registry, management calls



Executive components

- Object manager
 - Gestisce tutti gli Objects
 - Meccanismo comune ed uniforme per usare le risorse del sistema
 - protezione e limiti sull'uso (quote)
 - condivisione tra processi
 - traccia di qual oggetti sono in uso (garbage coll.)
- nessun oggetto al momento del boot

Struttura degli Objects (?)



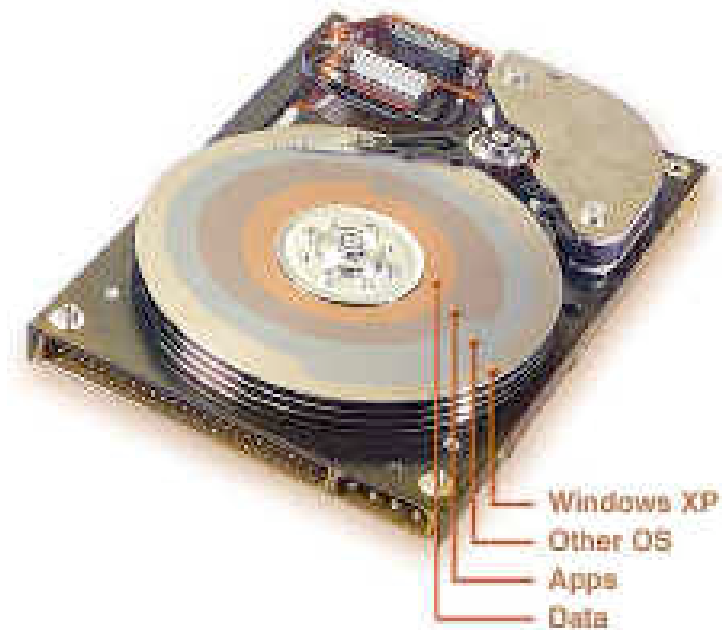
2 pool pagine, uno paged e l'altro pinned
user puo' creare oggetto: handle con puntatore a
oggetto, flag e diritti

Installazione Windows

- Partizioni
- Boot CD
 - (opzionale) Fare i dischi di avvio
- Scegliere varie opzioni
- Reboot (più di uno)

Partizioni

- Dimensionamento
- Tipi
- MBR
- Strumenti
 - Partizionare
 - Salvare/recuperare

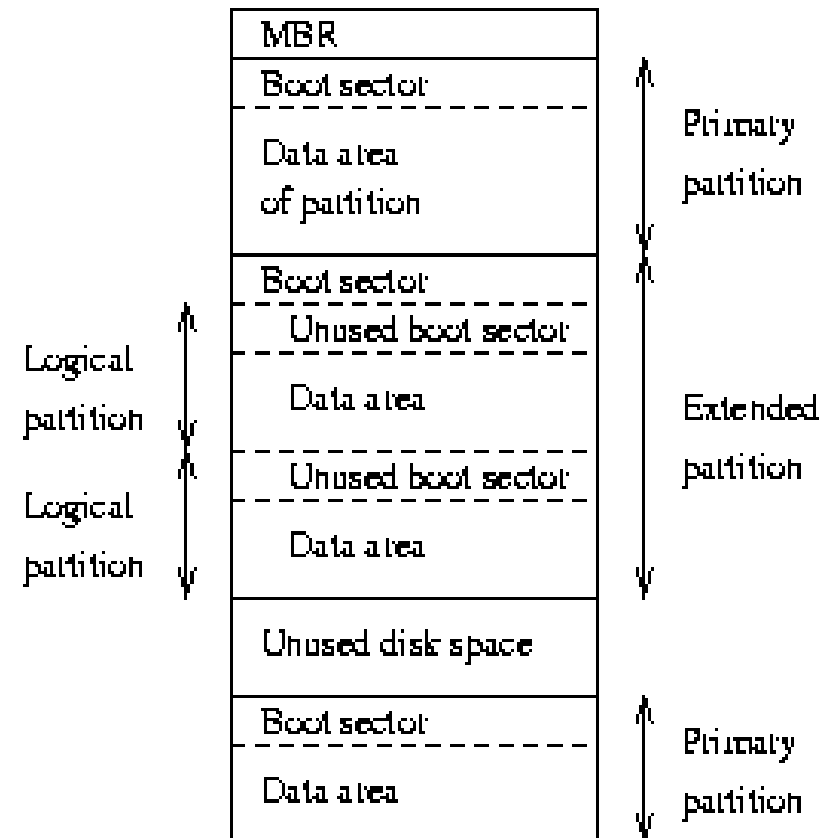


Dimensionamento

- Se si vuole installare anche Linux ☺ occorre dividere lo spazio su disco in più partizioni o avere più dischi
- Windows necessita circa 2GB (fino a 3/4 GB se si installano sw vari - ad es. l'ambiente di sviluppo)
- Linux, per una configurazione “normale”, varia da circa 1GB fino a 3/4 GB
 - Importante: per Linux servono due partizioni (una per il sistema e una per lo swap space)

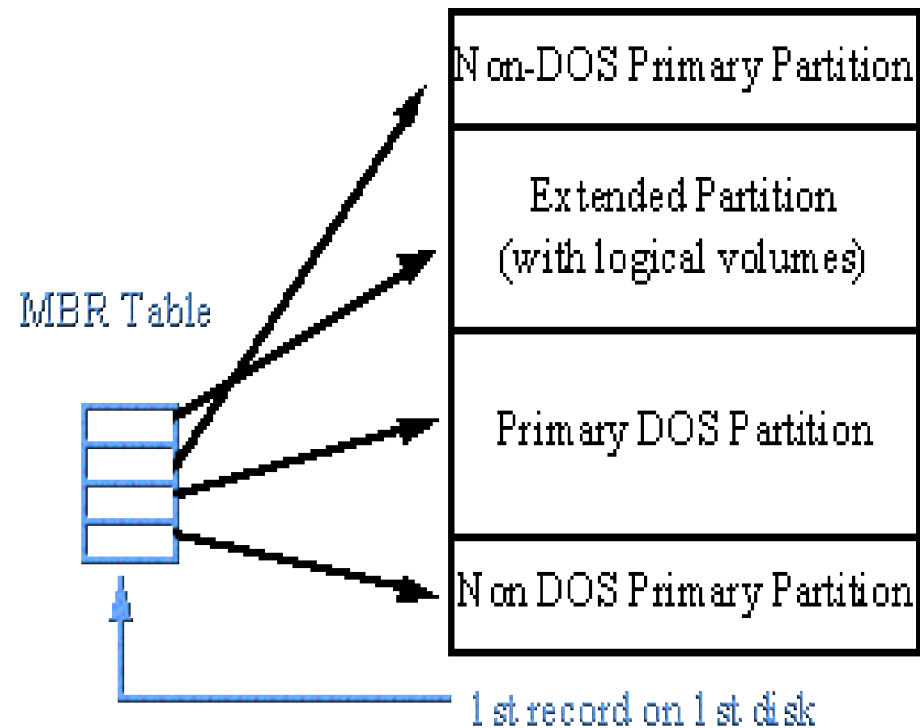
Tipi

- Riferendosi all'architettura di un PC standard:
 - Primary
 - Extended
 - Logical
- Dimensioni in settori, blocchi o cilindri
 - Begin, end



MBR (1)

- Master Boot Record
 - Tabella partizioni
 - Partizione attiva



MBR (2)

Master Boot Record

		Decimal Length in bytes
000	Primary Loader: Code to load and start boot sector of active partition	446
1BE	Partition 1	016
1CE	Partition 2	016
1DE	Partition 3	016
1EE	Partition 4	016
1FE	Magic number: 0xAA55	002
200		

MBR (3)

Partition Structure

000

Boot			
HD	SEC	CYL	
SYS			
HD	SEC	CYL	
Low		High	
Low		High	

010

Boot flag: 8 -> active

3D address start sector

System type code

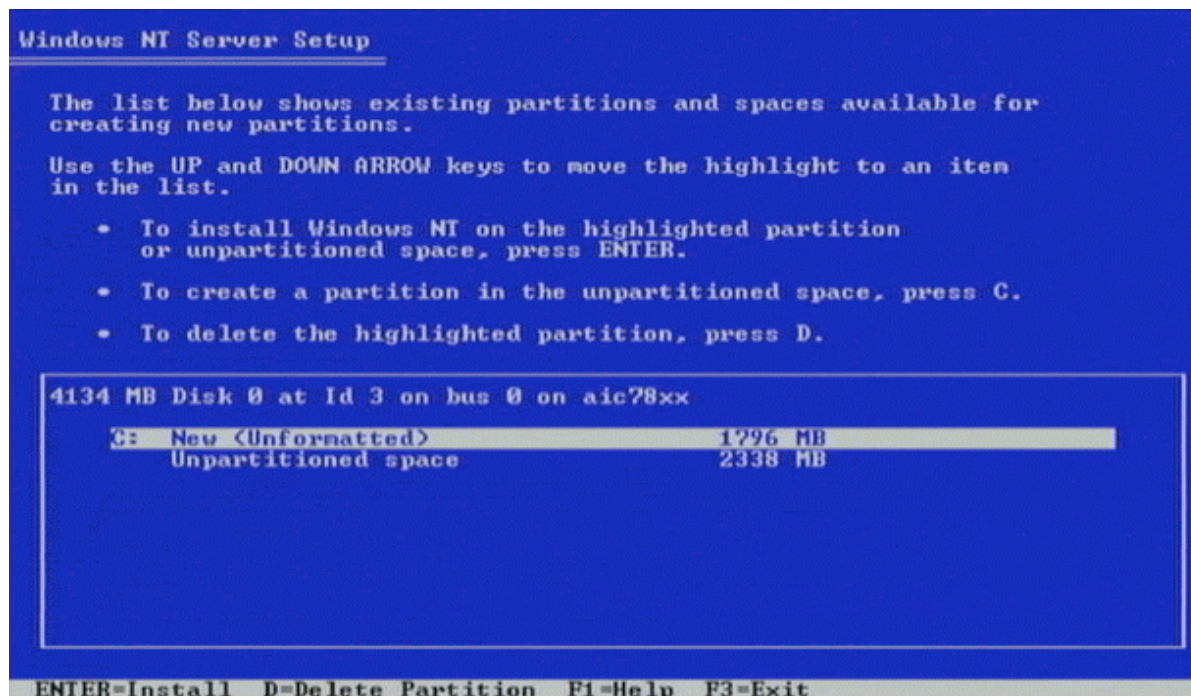
3D address end sector

Offset of start sector

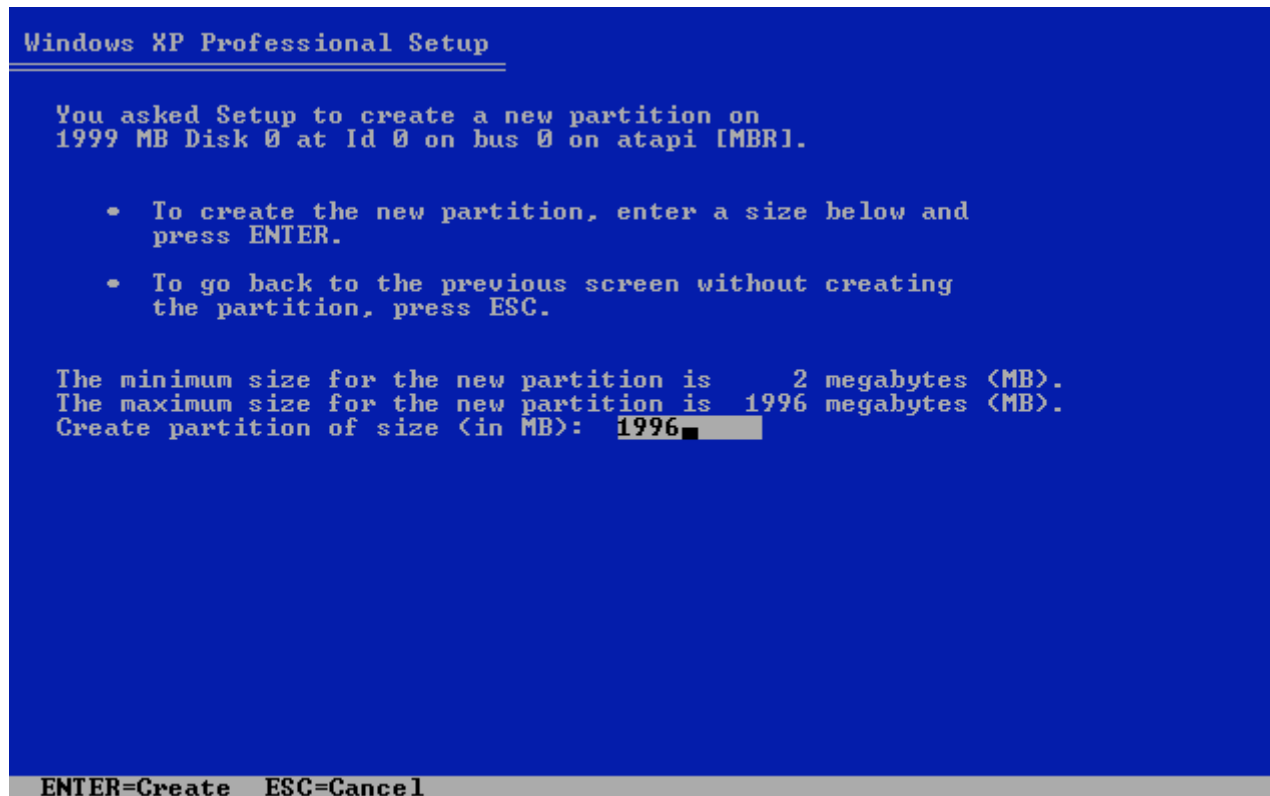
Number of sectors

Partizionare

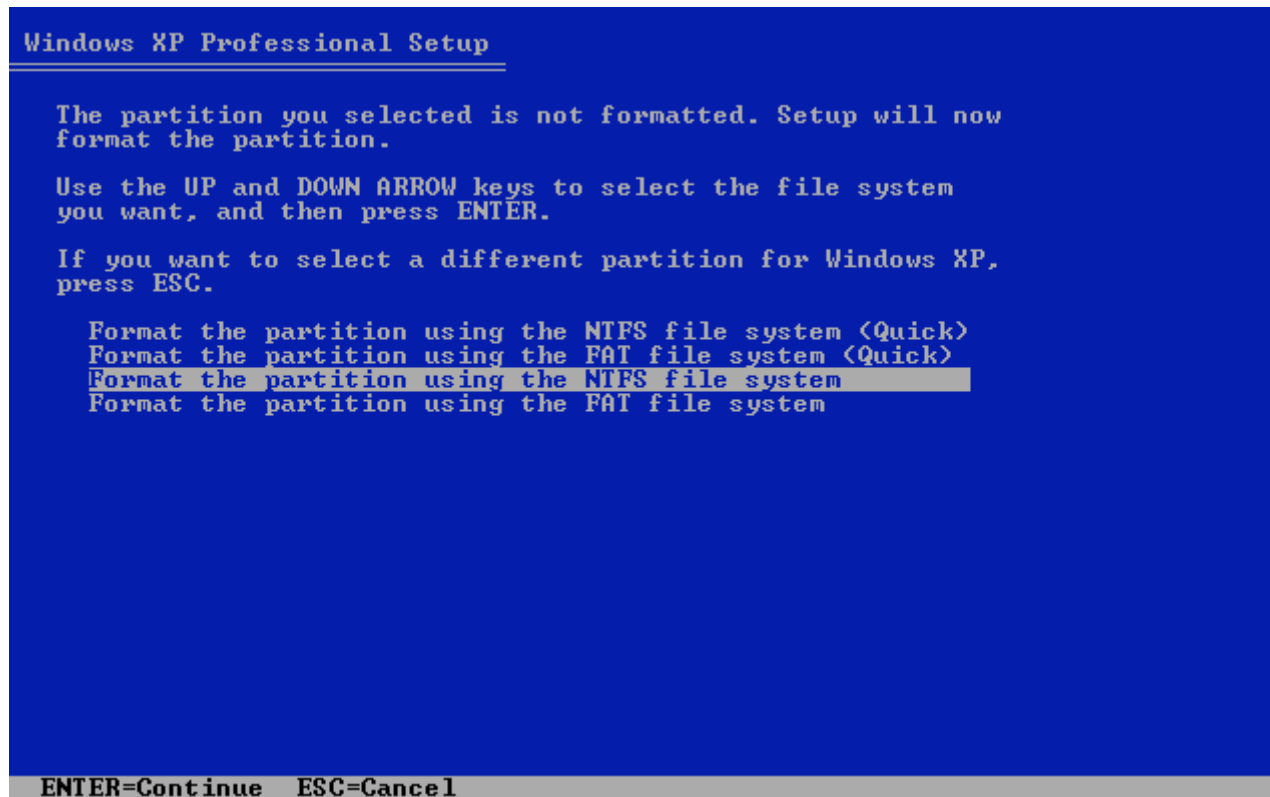
- Command prompt: Fdisk.exe
- All'installazione:



Partizionare (2)



Partizionare (3)



FDISK (1)

Your computer has a disk larger than 512 MB. This version of Windows includes improved support for large disks, resulting in more efficient use of disk space on large drives, and allowing disks over 2 GB to be formatted as a single drive.

IMPORTANT: If you enable large disk support and create any new drives on this disk, you will not be able to access the new drive(s) using other operating systems, including some versions of Windows 95 and Windows NT, as well as earlier versions of Windows and MS-DOS. In addition, disk utilities that were not designed explicitly for the FAT32 file system will not be able to work with this disk. If you need to access this disk with other operating systems or older disk utilities, do not enable large drive support.

Do you wish to enable large disk support (Y/N)? **[Y]**

FDISK (2)

- Create
- Scegliere tipo
- Specificare dimensione
- Uscire
- Reboot ☺

```
                Create DOS Partition or Logical DOS Drive

Current fixed disk drive: 1

Choose one of the following:

1. Create Primary DOS Partition
2. Create Extended DOS Partition
3. Create Logical DOS Drive(s) in the Extended DOS Partition

Enter choice: [ 1 ]
```

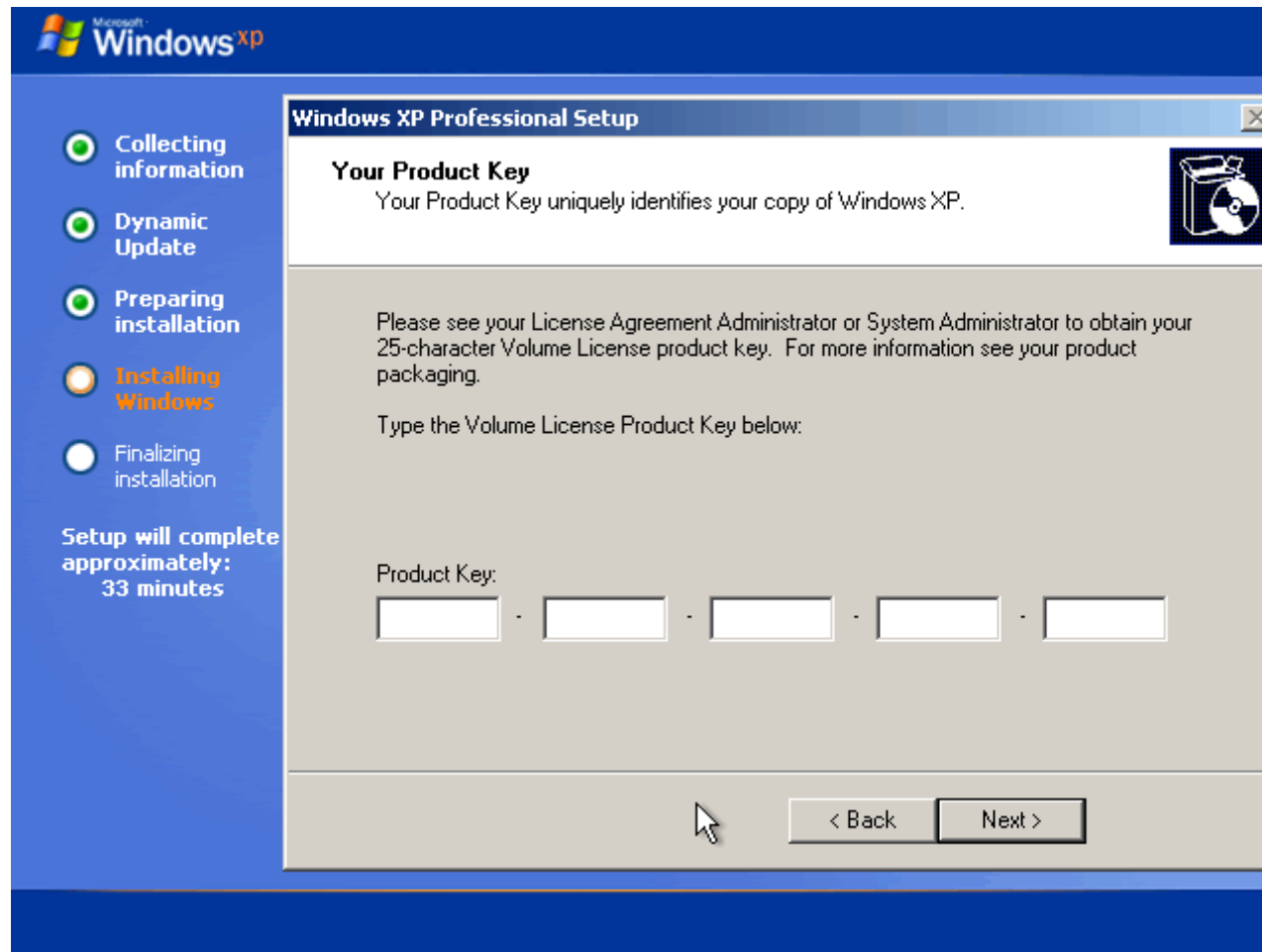
Boot da CD

- Inserire cd ☺
- Controllare BIOS settings

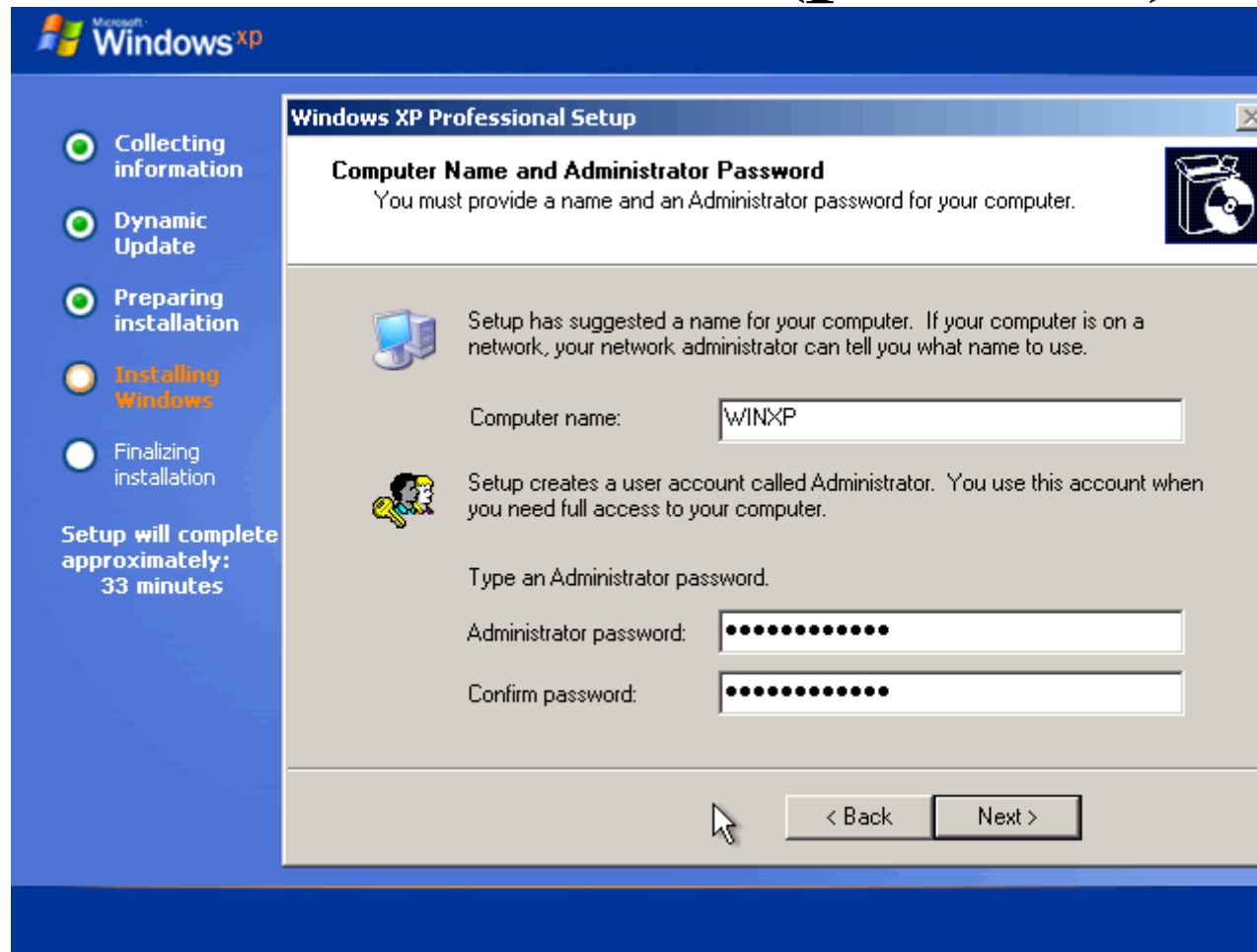
Boot da floppy

- E' possibile fare i floppy di boot usando un CD di installazione di Windows (NT/2000/XP) e un PC già installato
- Inserire CD ☺ e aprire un command prompt
- Andare in “winnt” (o simili)
- Eseguire “winnt /?” per avere l’elenco delle opzioni
- Trovare quella che fa scattare la creazione dei floppy di avvio

Punti salienti (license)



Punti salienti (passwd)



The screenshot shows the 'Windows XP Professional Setup' window. On the left, a blue sidebar contains a progress indicator with five steps: 'Collecting information', 'Dynamic Update', 'Preparing installation', 'Installing Windows' (highlighted in orange), and 'Finalizing installation'. Below the progress indicator, it states 'Setup will complete approximately: 33 minutes'. The main window has a title bar 'Windows XP Professional Setup' and a subtitle 'Computer Name and Administrator Password'. Below the subtitle, it says 'You must provide a name and an Administrator password for your computer.' The main area contains two sections. The first section, titled 'Computer name:', shows a suggested name 'WINXP' in a text box. The second section, titled 'Administrator password:', shows two text boxes for 'Administrator password:' and 'Confirm password:', both filled with dots. At the bottom right, there are '< Back' and 'Next >' buttons.

Microsoft Windows^{XP}

Windows XP Professional Setup

Computer Name and Administrator Password
You must provide a name and an Administrator password for your computer.

Setup has suggested a name for your computer. If your computer is on a network, your network administrator can tell you what name to use.

Computer name: WINXP

Setup creates a user account called Administrator. You use this account when you need full access to your computer.

Type an Administrator password.

Administrator password:

Confirm password:

< Back Next >

Installazione Ambiente Sviluppo

- Visual Studio
- (vedere doc PDF)

Installazione strumenti utili

- Performance monitor (già installato)
- TaskManager (già installato) ctrl-alt-del
- Resource Kit, SDK, Support Tools
 - PFMon, gflags, oh, qslice, pviewer, pstat, putlist, tlist, drivers, winobj, Pwalk, ...
 - Sono a linea di comando (quindi OK ☺)
- Dove pescarli?
 - www.microsoft.com, CD VisualStudio

Primo esempio

- Un classico: HelloWorld
- Solo per testare la correttezza dell'installazione
- Edit/Compile/execute

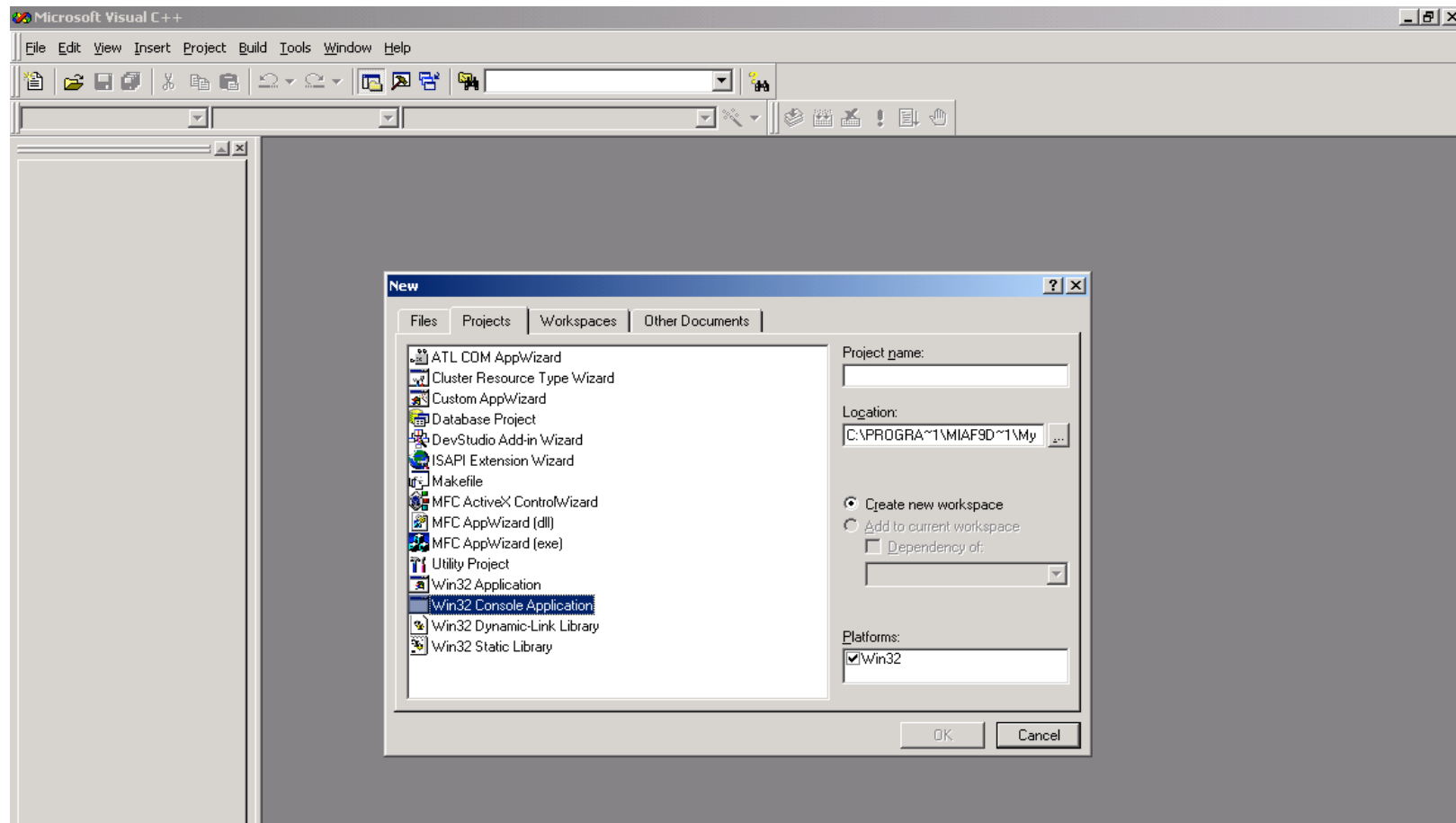
Sorgente

```
// hello.cpp : entry point for the console application.
```

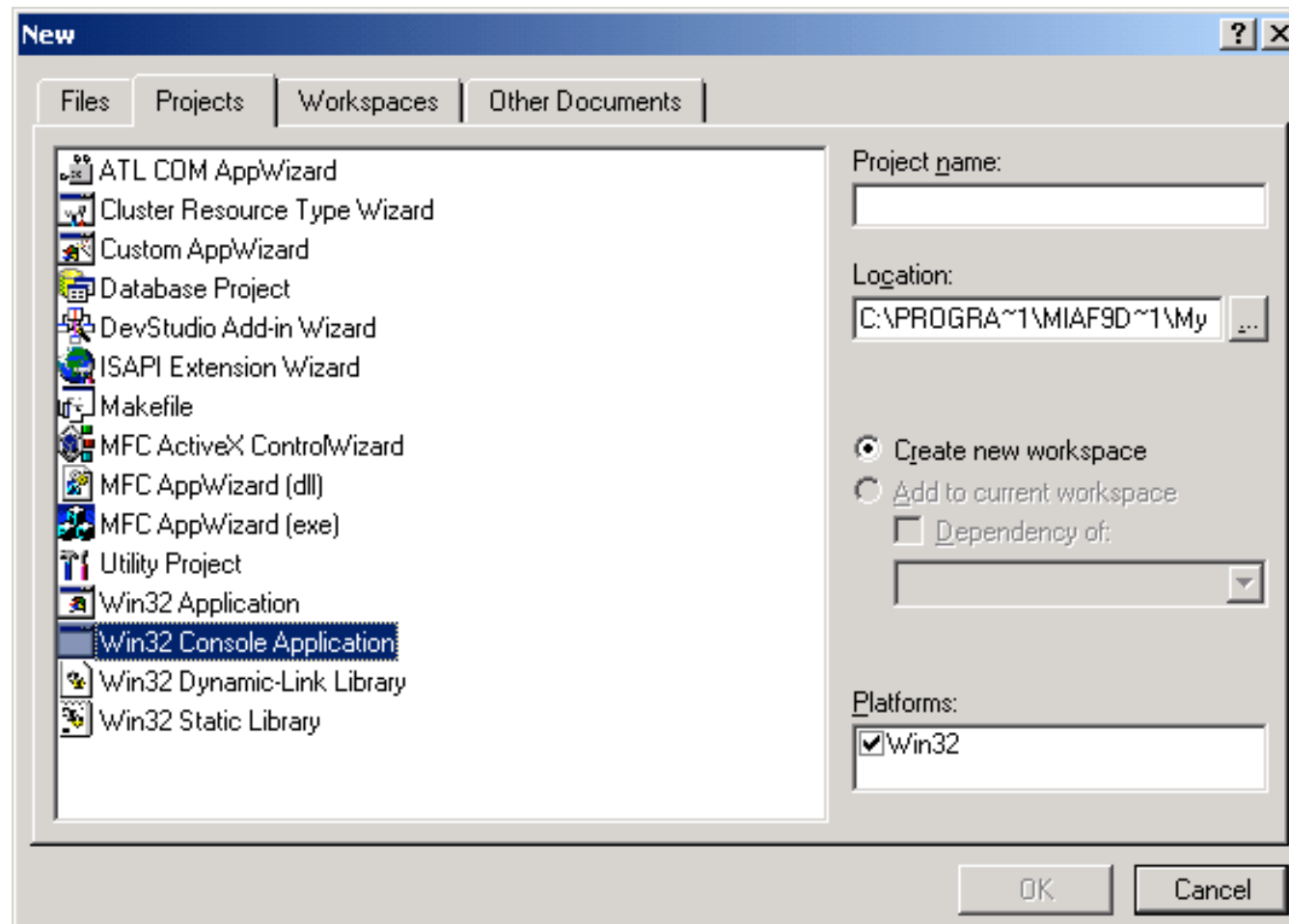
```
#include "stdafx.h"
```

```
int main(int argc, char* argv[])  
{  
    printf("Hello World!\n");  
    return 0;  
}
```

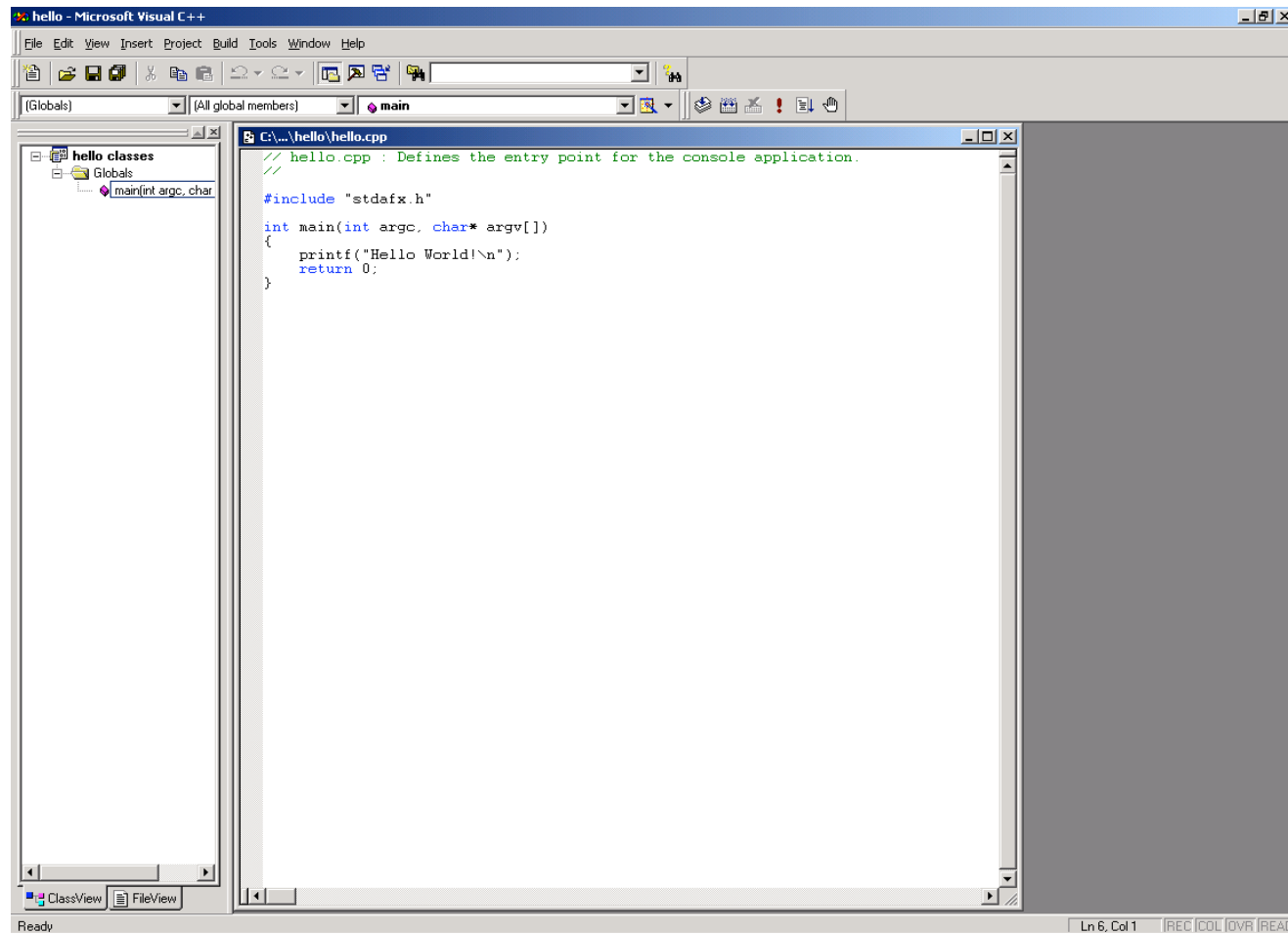
Tipo di progetto



Tipo di progetto (2)



Sorgente



Run

