Corso di Laurea Ingegneria Informatica

Laboratorio di Informatica

Capitolo 2

Il Sistema Operativo

Settembre 2006

Nota bene

- Alcune parti del presente materiale didattico sono derivate da:
  - trasparenze per il corso di Laboratorio di Informatica A.A. 2005-2006 prodotte da Alfonso Miola
- L'utilizzo di questo materiale è stato consentito dall'autore

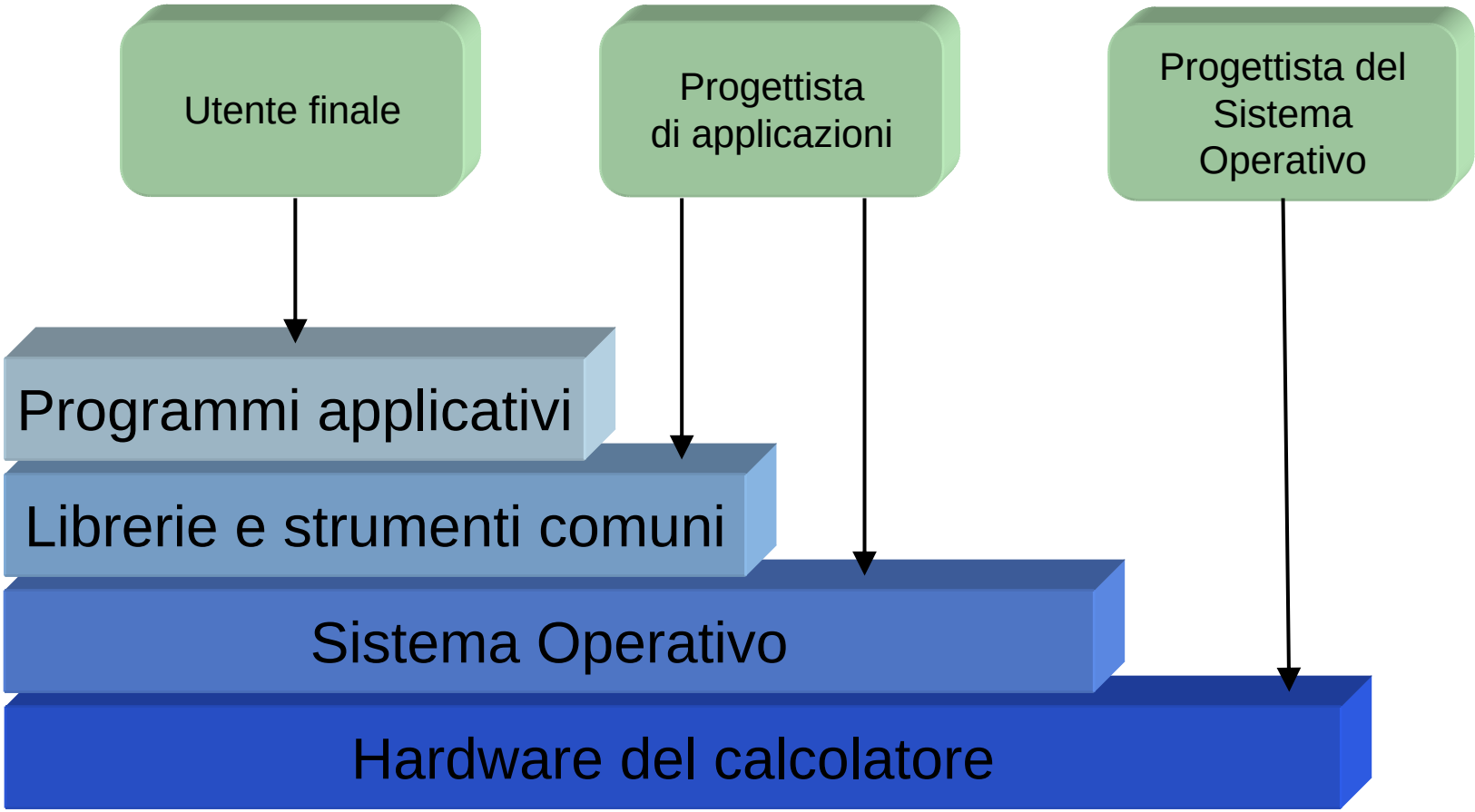
Contenuti

- L’architettura a strati di un calcolatore
- I compiti del sistema operativo
- L’esecuzione delle applicazioni ed il passaggio dei parametri

Cos’è il Sistema Operativo?

- Un programma che gestisce l’esecuzione dei programmi applicativi
- Un’interfaccia tra le applicazioni e l’hardware del calcolatore
- Un insieme di programmi che forniscono funzionalità di base

Architettura a strati di un calcolatore



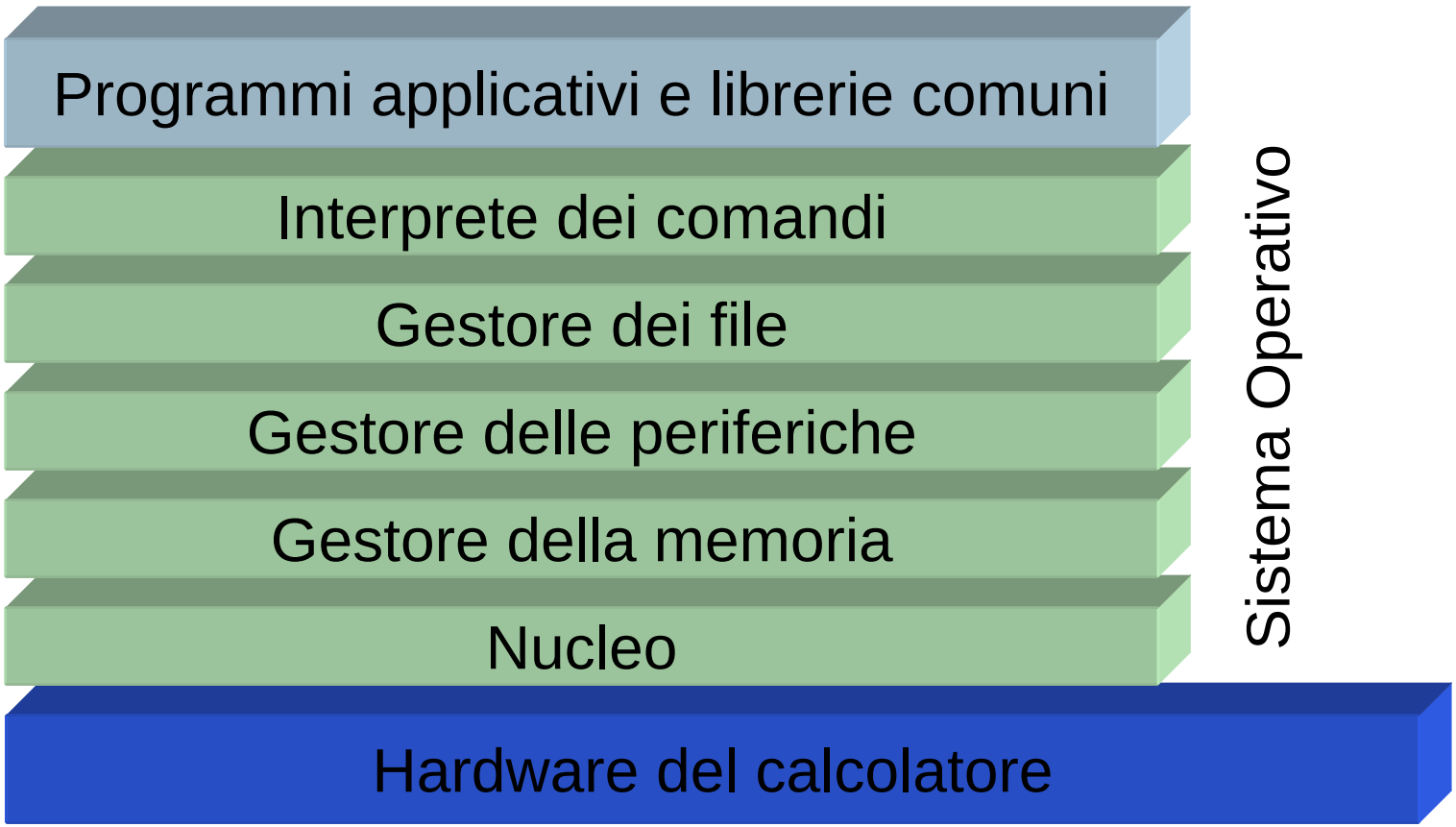
Il sistema operativo

- Il **sistema operativo** è il componente del software di base responsabile della gestione delle risorse del calcolatore, e in particolare di come le risorse vengono allocate alle applicazioni
  - ad esempio
    - l'utente può richiede al sistema operativo di eseguire una applicazione facendo doppio click sull'icona dell'applicazione stessa
    - l'utente può cancellare o rinominare un'applicazione
    - l'utente può esplorare i dischi fissi alla ricerca di una particolare applicazione

Architettura a livelli di un sistema operativo

- I sistemi operativi hanno una struttura complessa, che può essere descritta come una gerarchia di macchine virtuali
  - ciascun livello della gerarchia è relativo a una macchina virtuale del sistema operativo che gestisce una diversa tipologia di risorse
    - le risorse gestite sono i processori, la memoria, le periferiche, le memorie secondarie, l'interfaccia utente
  - l'utente del calcolatore interagisce solo con l'interprete comandi del sistema operativo

Architettura a livelli del sistema operativo

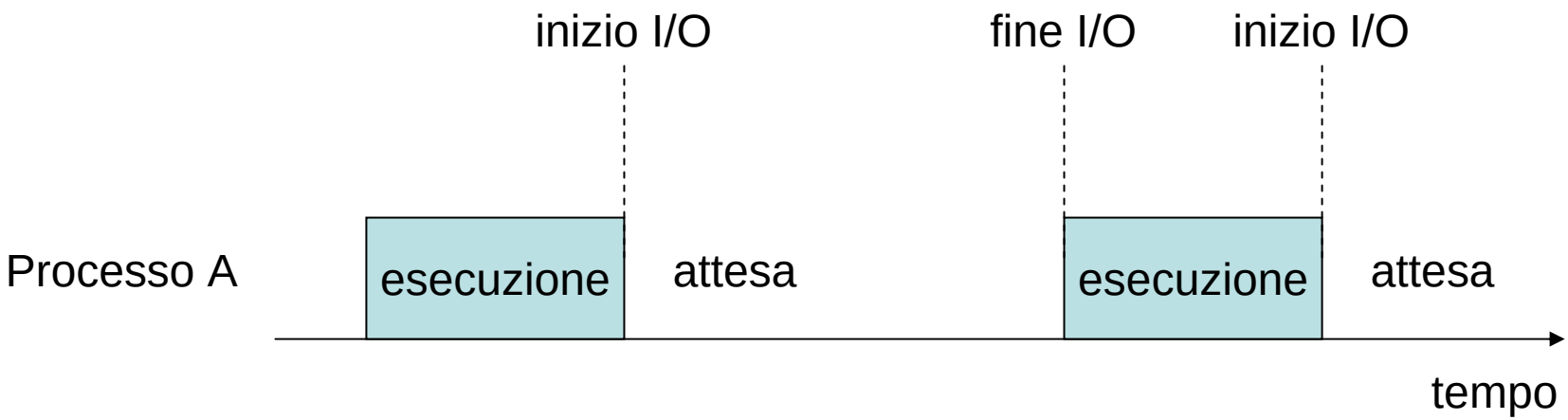


Nucleo

- Il **nucleo** (o **kernel**, o **gestore dei processi**) del sistema operativo è responsabile della gestione del processore
  - generalmente sotto il nucleo esiste fisicamente un solo processore
    - quindi il calcolatore potrebbe svolgere un solo programma alla volta (uniprogrammazione)
  - il nucleo ha il compito di ripartire il tempo di calcolo del processore ai programmi in esecuzione, realizzando la multiprogrammazione
    - ciascun programma ha l'impressione di essere l'unico programma eseguito dal calcolatore
  - il nucleo gestisce anche l'eventuale presenza di più processori

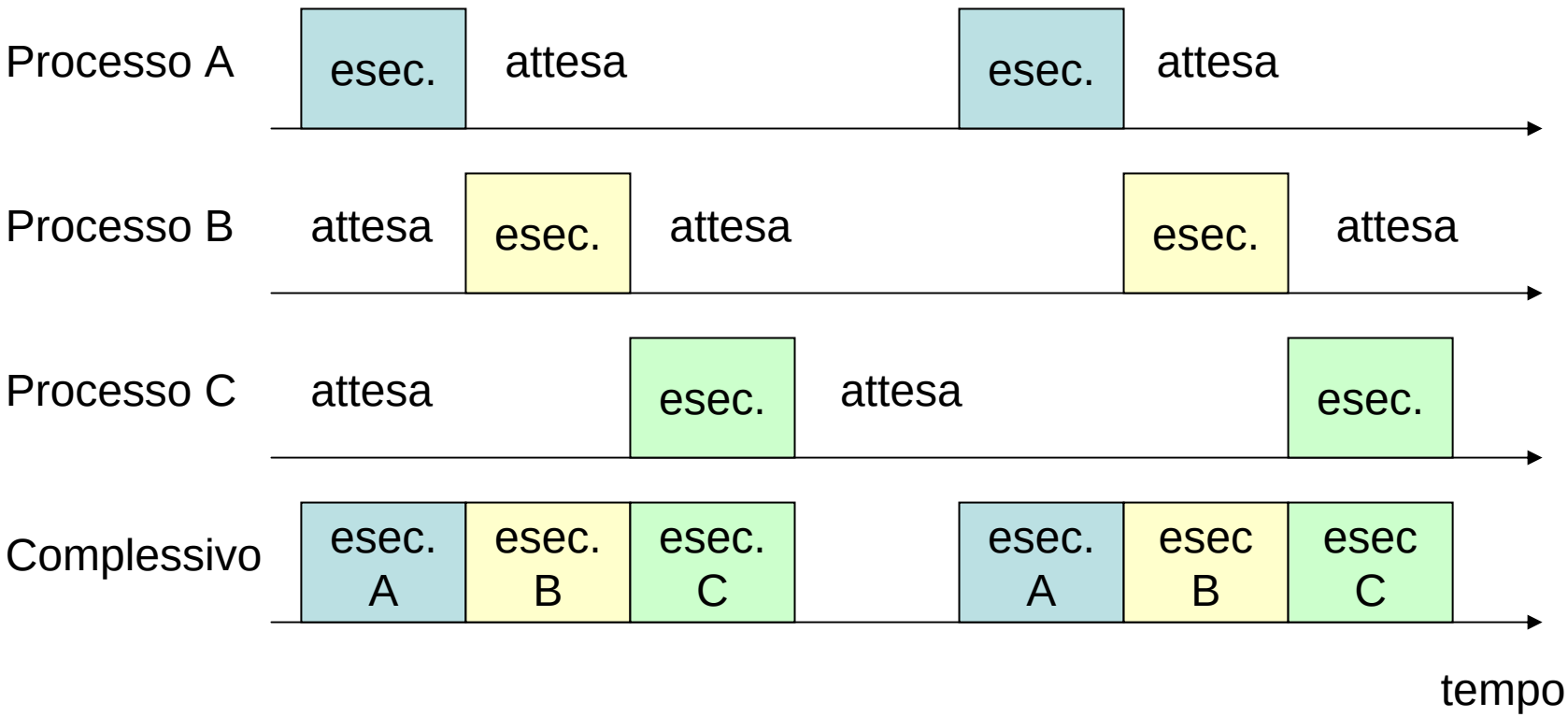
Uniprogrammazione

- Nella **uniprogrammazione** (ormai in disuso) il processore attende la fine delle operazioni di I/O per riprendere l'esecuzione dell'unico processo



Multiprogrammazione

- Nella **multiprogrammazione** o **multitasking** il processore salta da un processo all'altro



Gestore della memoria

- Il **gestore della memoria** consente l'allocazione dinamica della memoria centrale ai programmi in esecuzione
  - a ciascun programma viene allocata un'area di memoria virtuale sufficiente per la sua esecuzione
  - il gestore della memoria gestisce la corrispondenza tra le memorie virtuali e l'unica memoria reale
  - la dimensione della memoria virtuale può essere maggiore di quella reale
    - i dati possono essere parcheggiati temporaneamente nella memoria secondaria

Gestore delle periferiche

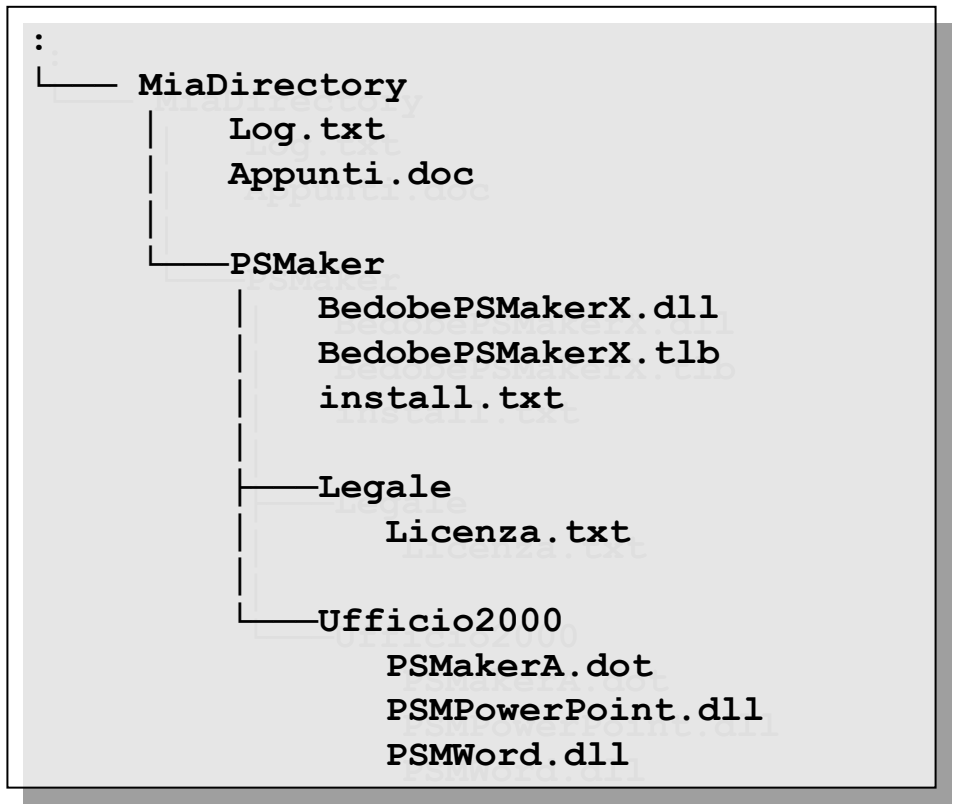
- Il **gestore delle periferiche** adatta la modalità d'uso delle singole periferiche (che possono essere estremamente diverse) a quello di poche tipologie di periferiche virtuali
  - ad esempio, un programma può usare una stampante senza conoscere i dettagli precisi di funzionamento della stampante fisicamente disponibile
  - un **driver** è un adattatore di dati dal formato virtuale (generico) usato sopra il gestore delle periferiche a quello reale della periferica effettivamente in uso

Il file system

- Il **file system** gestisce le informazioni memorizzate nelle memorie di massa (dischi)
  - le informazioni vengono distribuite in file, cartelle e volumi
    - un **file** è una sequenza di lunghezza variabile di byte e costituisce l'unità di dati elementare gestita dal file system
    - una **cartella** (o **directory**, o **direttorio**) è un contenitore di file e cartelle
    - un **volume** è una unità logica che contiene file e cartelle e corrisponde ad una porzione (detta **partizione**) della memoria di massa gestita dal file system

Alberi di cartelle

- Le cartelle, potendo contenere altre cartelle, sono organizzate gerarchicamente ad albero



Volumi e cartelle

- I sistemi Windows gestiscono un albero di cartelle separato per ogni volume
  - questa caratteristica deriva dal DOS “Disk Operating System”
  - ad esempio:
    - il file **`a:\miadir\prova.txt`** risiede nel volume **`a:`**
    - il file **`c:\win\log.txt`** risiede nel volume **`c:`**
- I sistemi Unix (come per esempio Linux) presentano all’utente un singolo albero di cartelle, nascondendo l’effettiva dislocazione dei file nei volumi
  - ad esempio
    - il file **`/home/user/miadir/prova.txt`** potrebbe risiedere su un disco diverso da quello su cui risiede il file **`/usr/bin/ls`**, ma entrambi i file appartengono allo stesso albero di cartelle

File

- Un **file** è l'unità di dati elementare gestita dal file system ed è caratterizzato da
  - un contenuto, cioè una sequenza di byte
  - un identificatore, unico per ogni file
- I dati contenuti nel file devono essere opportunamente interpretati
  - le regole con cui interpretare il file sono chiamate **formato** del file
  - i formati dei file corrispondono a codifiche convenzionali di informazioni adottatte dalle applicazioni

Nome dei file

- Un file è identificato da un **percorso** ed un **nome**
  - il **percorso**
    - indica la sequenza delle cartelle che occorre traversare per raggiungere il file
    - in alcuni sistemi indica anche il volume in cui si trova il file
  - il **nome**
    - unico all'interno della cartella in cui si trova il file
    - il nome può contenere il carattere "." (punto)
    - la parte terminale del nome dopo l'ultimo punto (quando presente) viene chiamata **estensione**
      - l'estensione è lunga generalmente tre caratteri (esempi: .txt .doc .exe .htm, …)
      - in alcuni sistemi (tipicamente DOS e Windows) l'estensione identifica il formato del file
- Per esempio, il nome d:\Home\Java\Dispense\01_calcolatore.ppt
  - d:\Home\Java\Dispense\ è il percorso
  - 01_calcolatore.ppt è il nome del file
  - ppt è l'estensione e suggerisce che il file possa essere interpretato correttamente dall'applicazione PowerPoint

Operazioni sui file

- Il sistema operativo mette a disposizione degli utenti le seguenti operazioni
  - **creazione**
    - l'utente, o più spesso un'applicazione lanciata dall'utente, può creare file ed aggiungerli al file system
  - **cancellazione**
    - l'utente può cancellare un file
  - **apertura**
    - l'utente, o più spesso un'applicazione lanciata dall'utente, può dichiarare di essere interessato a leggere o scrivere su un file
    - questa dichiarazione serve a prevenire la modifica del file da parte di altri utenti/applicazioni
  - **chiusura**
    - l'utente dichiara di aver terminato la lettura/modifica del file
  - **lettura**
    - l'utente legge il file
  - **scrittura**
    - l'utente aggiorna il file

Rapporto tra file e applicazioni: file eseguibili

- Alcuni file, chiamati **programmi** o **applicazioni**, sono riconosciuti dal sistema operativo come eseguibili
  - nei sistemi Windows i file eseguibili sono determinati dalla loro estensione (.exe .com .bat)
  - nei sistemi Linux i file eseguibili non si possono riconoscere dal nome, ma sono esplicitamente etichettati come tali con speciali comandi

Rapporto tra file e applicazioni: file dati

- Il sistema operativo può gestire anche l'associazione tra file e applicazioni in grado di interpretare il loro formato
- Nei sistemi Windows
  - ogni estensione (.doc, .txt, .ppt, .pdf, ecc) viene associata ad una applicazione in grado di interpretare il formato del file (rispettivamente Winword, Notepad, PowerPoint, AcrobatReader, ecc)
  - quando si fa doppio click su un file, viene eseguita l'applicazione associata all'estensione del file, e il file selezionato viene aperto automaticamente dall'applicazione
- In alcuni sistemi Unix
  - l'applicazione in grado di interpretare il formato del file viene desunta dalla lettura dei primi byte del file stesso

Esercizio

- Su una piattaforma Windows
  1. considera un file con estensione .pdf
  2. rinomina il file con estensione .doc
  3. fai doppio-click sul file
     - cosa ti aspetti che succeda?
     - perché?

- Ripeti lo stesso esperimento su una piattaforma Unix (per esempio Linux)
  - cosa ti aspetti che succeda?
  - perché?

Programma e processo

- Per un file eseguibile o **programma**, oltre alle consuete operazioni (creazione, cancellazione, apertura, chiusura, lettura e scrittura) l'utente può richiedere l'operazione di **esecuzione**
- Quando esegue un programma il sistema operativo
  - reperisce nel file system il file contente il codice eseguibile del programma
  - alloca al programma le risorse necessarie per la sua esecuzione (ad esempio, una certa quantità di memoria centrale)
  - copia il codice eseguibile del programma in memoria centrale
  - avvia l'esecuzione del programma
- Un programma in esecuzione viene chiamato **processo**
  - il processo non va confuso con il file eseguibile, o programma, corrispondente
  - l'utente potrebbe richiedere l'esecuzione contemporanea dello stesso programma più volte, generando diversi processi tutti corrispondenti allo stesso programma

Interprete comandi

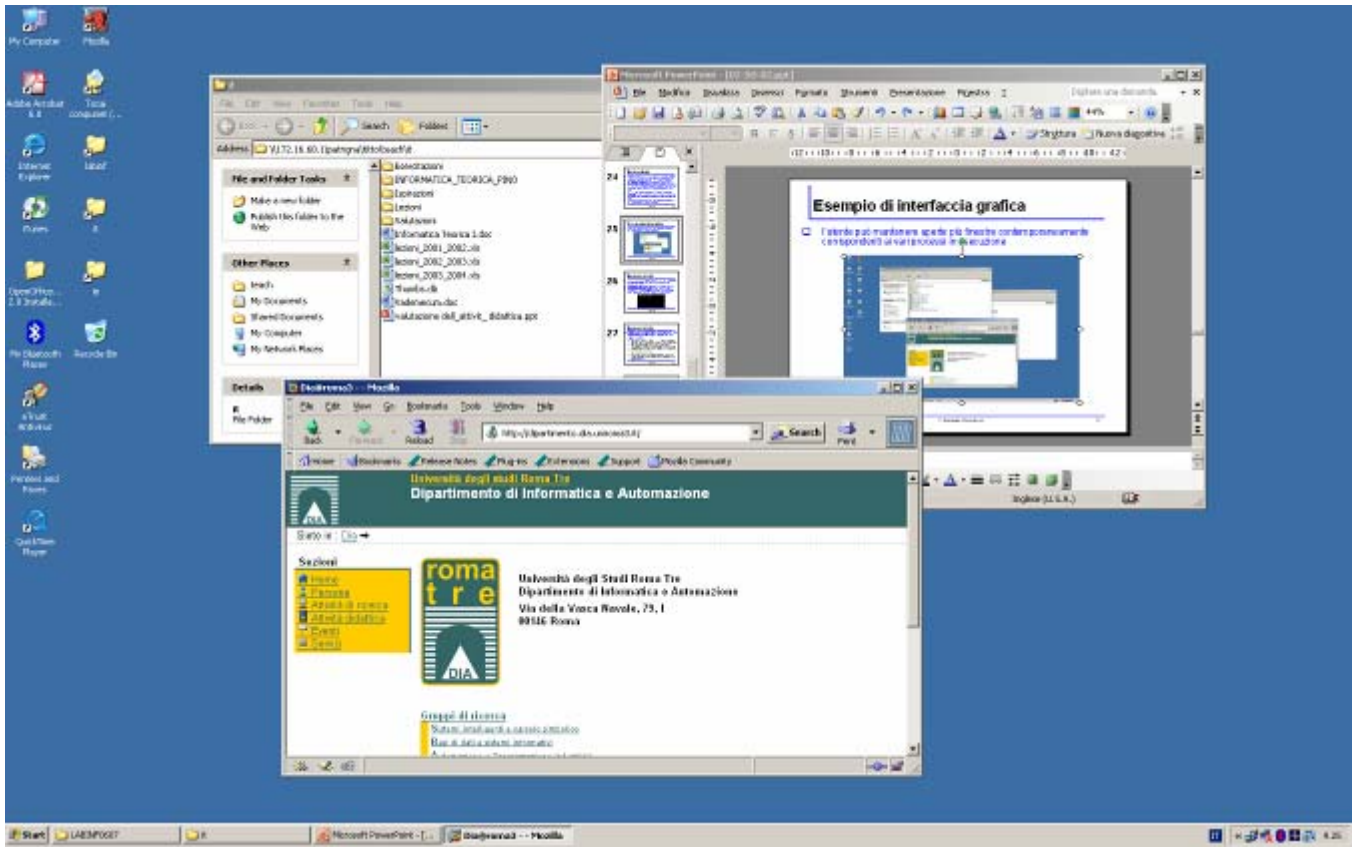
- L'**interprete comandi** è l'interfaccia utente del sistema operativo
  - definisce le operazioni che possono essere utilizzate direttamente dall'utente finale, chiamate **comandi**
  - esempi di comandi sono
    - esecuzione di una applicazione
    - apertura di un documento (eseguendo l'applicazione in grado di interpretare il suo formato)
  - l'interprete comandi può essere
    - basato su una **interfaccia a caratteri** — come nei sistemi operativi Unix, ma anche nel "prompt dei comandi" di Windows
    - basato su una **interfaccia grafica** — come nei sistemi operativi Windows e nell'ambiente X Window di Unix

Interfaccia grafica

- l'elemento principale è la **scrivania** (o **desktop**) sulla quale riposano le cartelle ed i file che corrispondono ai dati presenti sul disco fisso
- sul desktop si possono aprire delle **finestre** che astrattamente corrispondono ai documenti che materialmente possono essere disposti su una scrivania
- cartelle e finestre sono oggetti che l'utente può manipolare tramite il **mouse**: drag & drop, menu contestuale, point&click, doppio click, ecc.
- sul desktop appaiono anche i programmi disponibili sotto forma di icona
- un **menù** che consente di selezionare delle voci corrispondenti a programmi
- l'utente può richiedere l'esecuzione di un programma facendo doppio click sull'icona corrispondente o selezionando la voce opportuna del menù
- l'utente può mantenere aperte più finestre contemporaneamente corrispondenti ai vari processi in esecuzione

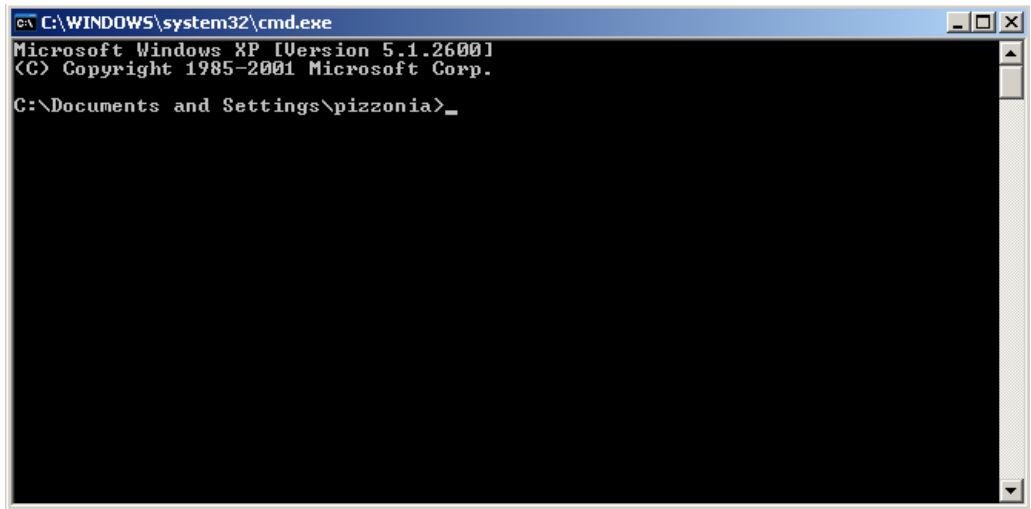
Esempio di interfaccia grafica

- l'utente può mantenere aperte più finestre contemporaneamente corrispondenti ai vari processi in esecuzione



Interfaccia a caratteri

- L'**interfaccia a caratteri** (detta anche **shell dei comandi**) consente di interagire con il sistema operativo scrivendo e leggendo delle sequenze di caratteri
- Nei sistemi Unix l'interfaccia a caratteri è lo strumento principe di interazione con il sistema operativo
- Nei sistemi Windows si può ottenere un'interfaccia a caratteri selezionando "Programmi" → "Accessori" → "Prompt dei comandi" oppure selezionando "Avvio" → "Esegui…" e digitando "cmd"



Interfaccia a caratteri

- Utilizzando una interfaccia a caratteri l'utente può richiedere l'esecuzione di un programma nel seguente modo
  - una speciale stringa, detta **prompt**, segnala all'utente che l'interfaccia a caratteri è pronta a ricevere comandi
  - l'utente scrive il nome del file corrispondente al programma da eseguire e batte il tasto **invio** (o **return**)
  - il sistema operativo avvia un processo corrispondente al programma selezionato
  - quando il processo è terminato all'utente riappare il prompt dei comandi che gli consente di immettere un nuovo comando

Parametri sulla linea di comando

- L'interfaccia a caratteri consente all'utente di specificare delle stringhe aggiuntive (dette **parametri**) in coda al nome del programma o del comando da eseguire
  - Esempi
    - `dir c:\miadirectory`
    - `cd ..`
    - `help dir`
    - `winword miofile.doc`
    - `dir *.exe *.com`
- I progettisti di applicazioni hanno a disposizione dei costrutti che consentono all'applicazione di recuperare la lista delle stringhe (separate da spazi) presenti sulla linea di comando

Flag

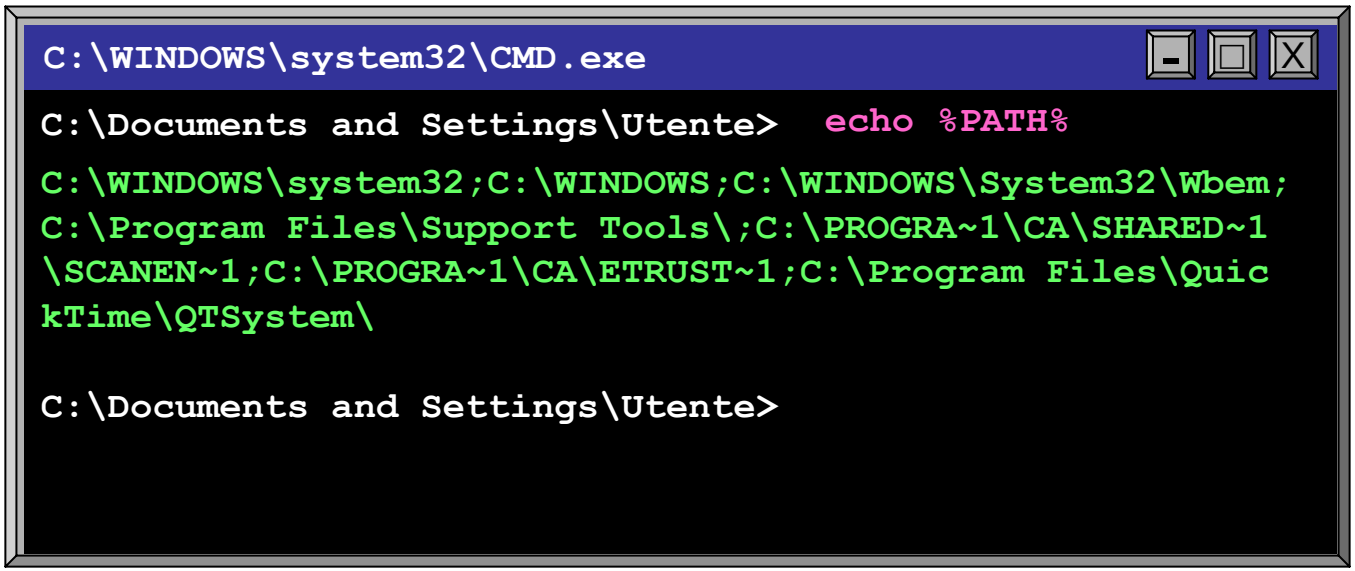
- I parametri vengono chiamati **flag** quando sono brevi stringhe precedute da caratteri di controllo
  - Esempi
    - `dir /a`
    - `cd /?`
- Alcune applicazioni richiedono flag seguiti da argomenti accessori
  - Esempi
    - `tar -t -v -z -f miofile.tgz miadir`
    - `javac -g -source 1.3 -encoding UTF8 file.java`

Variabili d’ambiente

- Le **variabili d’ambiente** sono un insieme di stringhe associate a dei nomi
- Esempi di variabili d’ambiente
  - HOMEDRIVE=C:
    - al nome “HOMEDRIVE” è associata la stringa “C:”
  - OS=Windows_NT
    - al nome “OS” è associata la stringa “Windows_NT”
  - TEMP=C:\tmpdir
    - al nome “TEMP” è associata la stringa “C:\tmpdir”
- I progettisti delle applicazioni e dei comandi hanno a disposizione dei costrutti che consentono all’applicazione di recuperare la stringa associata ad uno specifico nome (oppure tutte le coppie nome=valore)

Vedere il valore di una variabile d'ambiente

- L'utente può chiedere all'interfaccia a caratteri il valore della stringa associata ad una variabile d'ambiente
  - su una piattaforma Windows occorre eseguire:
    `echo %NOMEVARIABILE%`
  - su una piattaforma Unix/Linux occorre eseguire:
    `echo $NOMEVARIABILE`
    oppure **`echo ${NOMEVARIABILE}`**

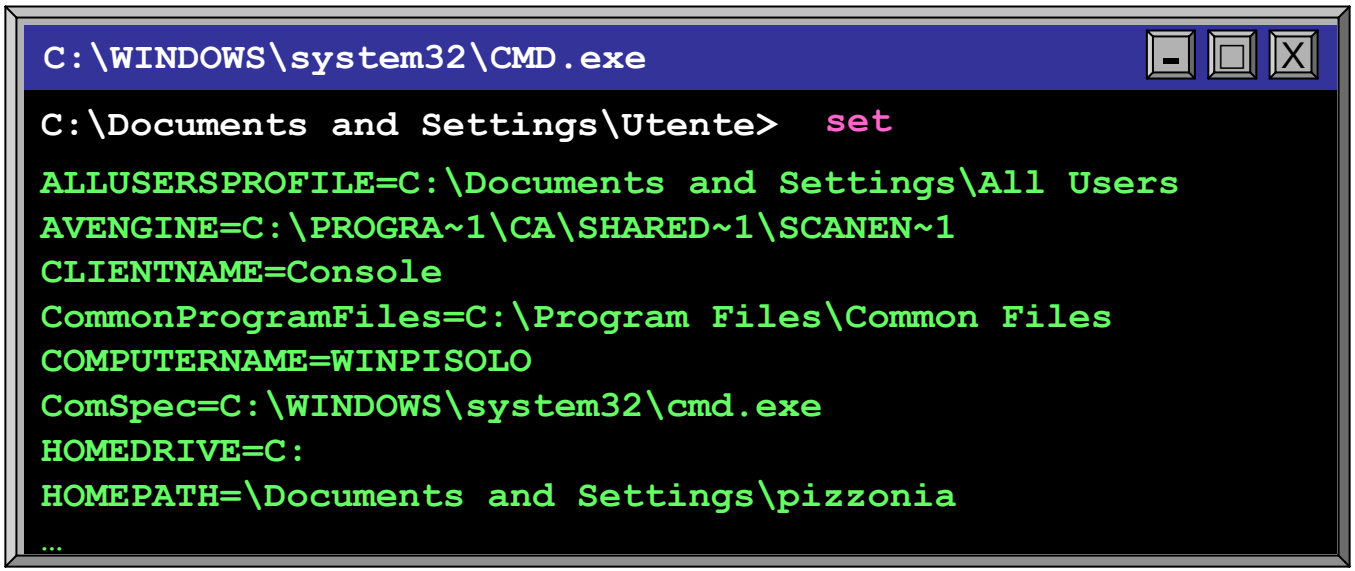


Vedere il valore di tutte le variabile d'ambiente

- L'utente può chiedere all'interfaccia a caratteri il valore di tutte le variabili d'ambiente
  - su una piattaforma Windows occorre eseguire:

    `set`
  - su una piattaforma Unix/Linux occorre eseguire:

    `env`

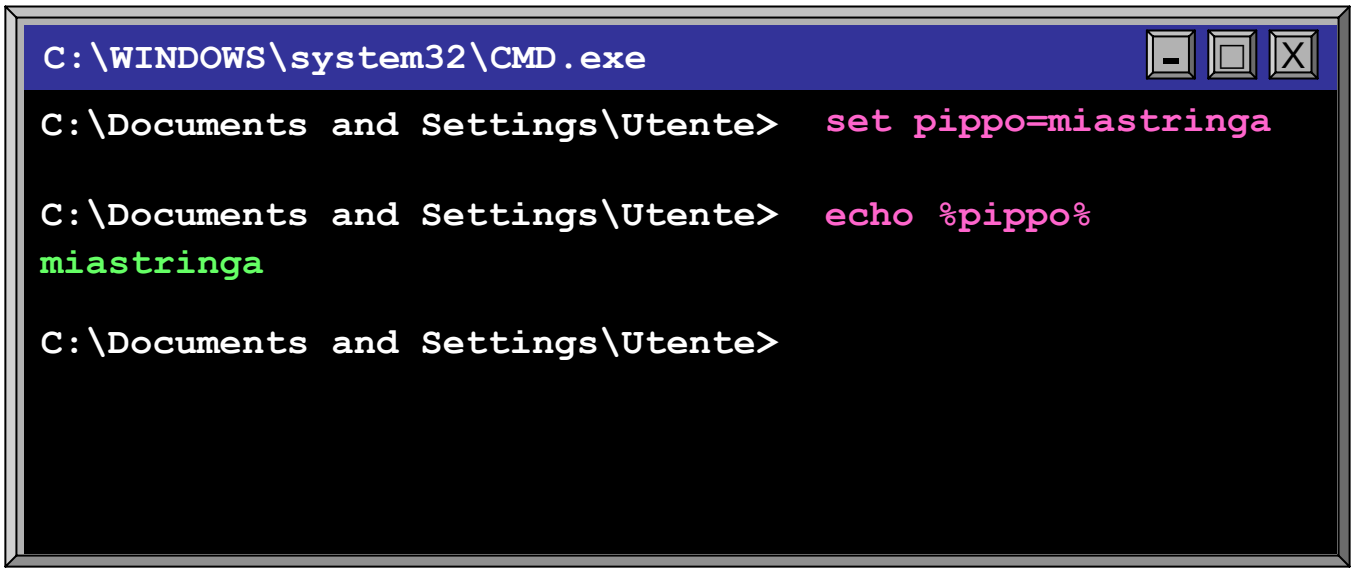


Assegnare una stringa ad una variabile d'ambiente

- L'utente può assegnare una stringa ad una variabile d'ambiente con i seguenti comandi
  - su una piattaforma Window:

    `set pippo=mia stringa`
  - su una piattaforma Unix/Linux:

    `pippo=mia stringa`

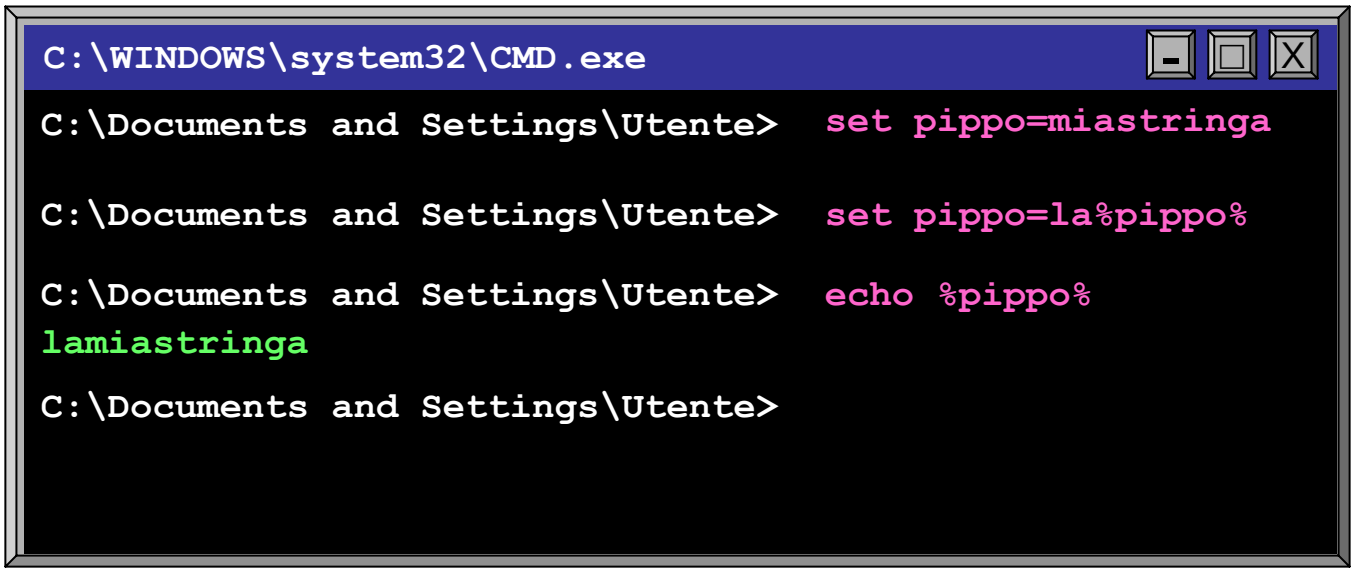


Aggiornare una variabile d'ambiente

- L'utente può aggiornare il valore di una variabile d'ambiente
  - su una piattaforma Window:

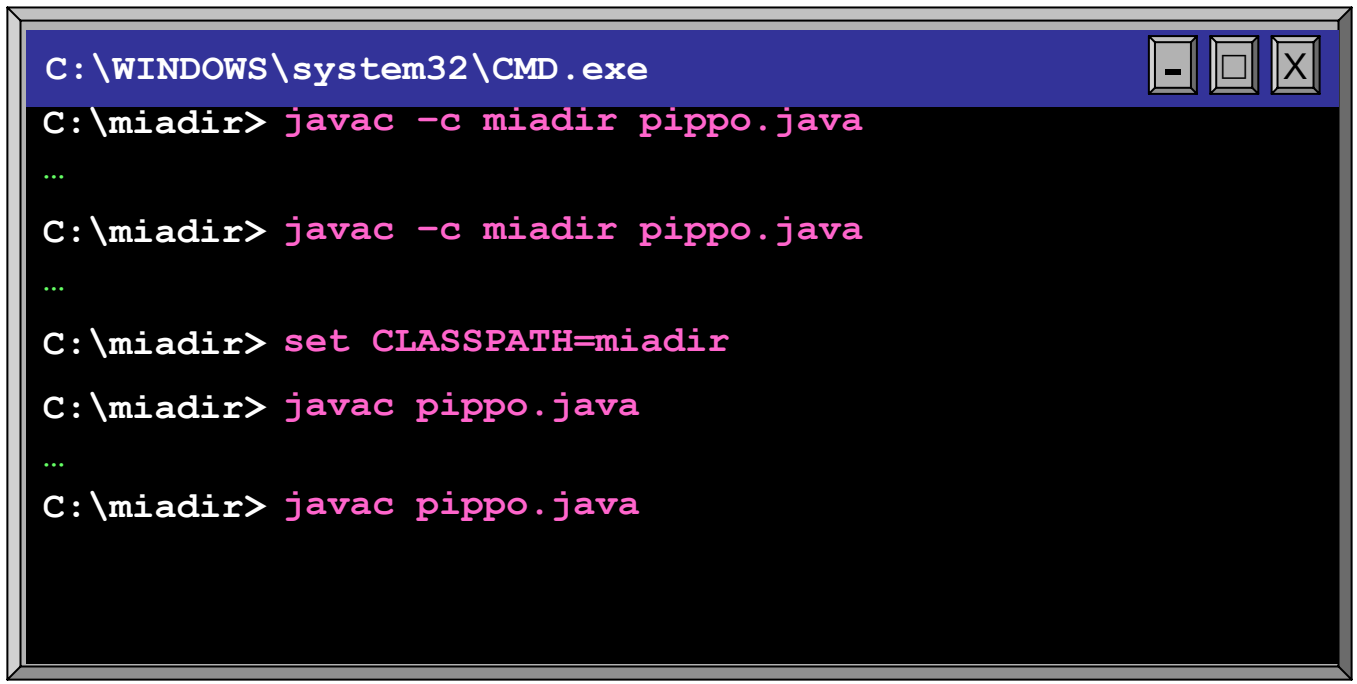
    `set pippo=la%pippo%`
  - su una piattaforma Unix/Linux:

    `pippo=la${pippo}`



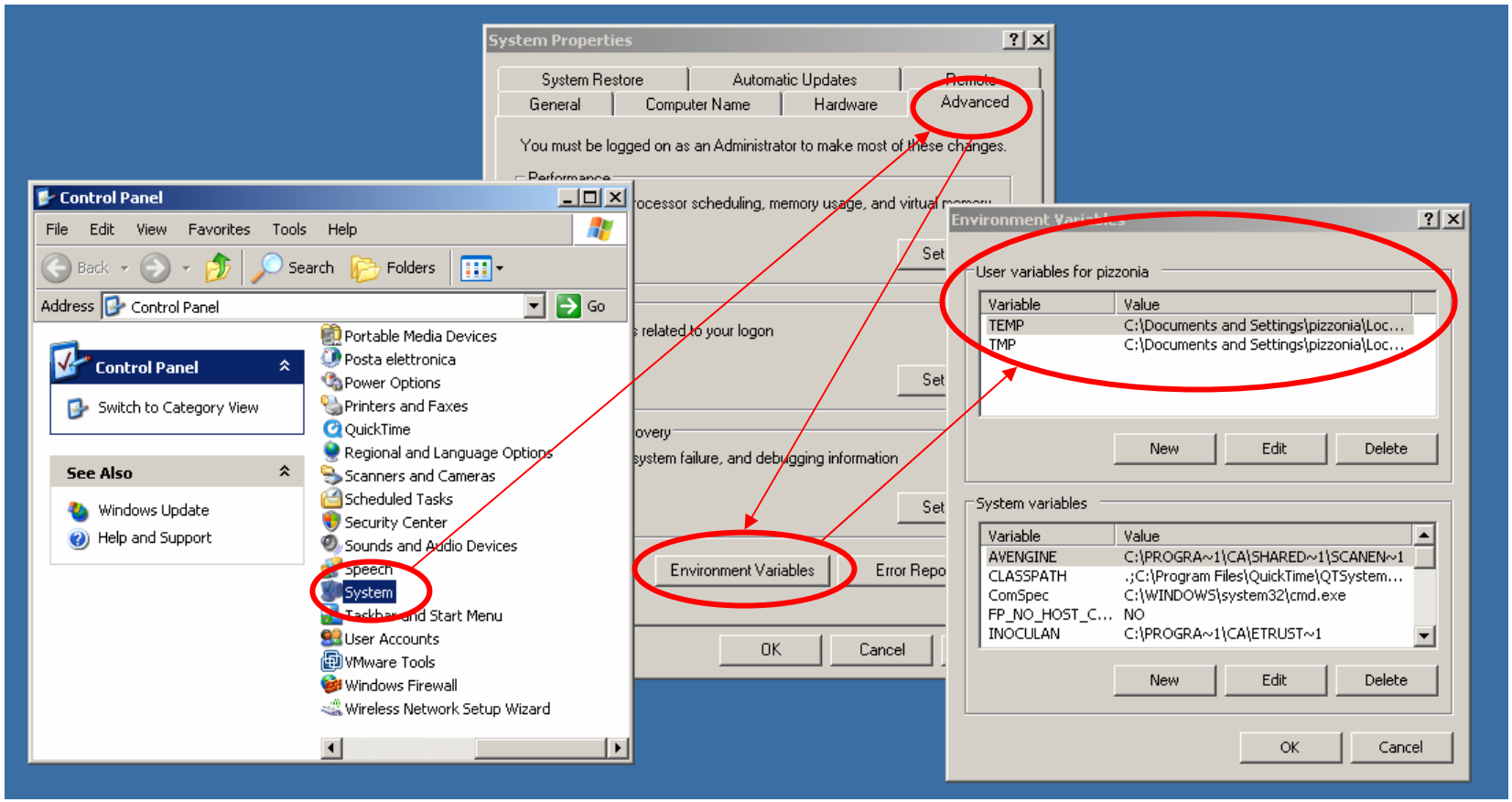
Parametri sulla linea di comando e variabili d'ambiente

- Alcune applicazioni accettano alcune opzioni sia come parametri sulla linea di comando che come variabili d'ambiente
- In questo caso, se il comando deve essere ripetuto spesso con gli stessi parametri, è opportuno settare la corrispondente variabile d'ambiente



Variabili d'ambiente

- E' possibile definire variabili d'ambiente per tutte le interfacce a caratteri che saranno in futuro create sulla macchina



Rifermimenti al libro di testo

- Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al capitolo 1 sull'architettura dei calcolatori:
  - 1.4 Il sistema operativo
    - 1.4.1 Architettura di un sistema operativo