LABORATORIO DI PROGRAMMAZIONE
Corso di laurea in matematica

6 – I SISTEMI OPERATIVI

Marco Lapegna
Dipartimento di Matematica e Applicazioni
Universita' degli Studi di Napoli Federico II

wpage.unina.it/lapegna

Dalla precedente lezione

Il linguaggi di programmazione sono lo strumento per tradurre algoritmi in programmi

Evoluzione dei linguaggi di programmazione

- Linguaggio macchina
- Linguaggio assembly
- Linguaggio ad alto livello

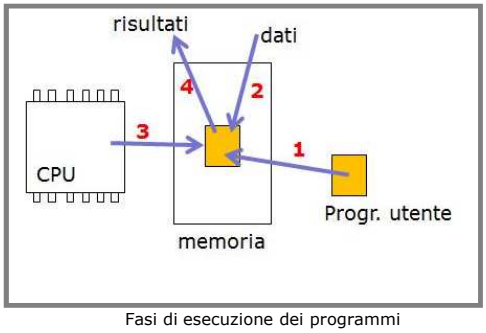
Ma come viene gestita l'esecuzione di un programma?

I Sistemi Operativi

Esecuzione dei programmi

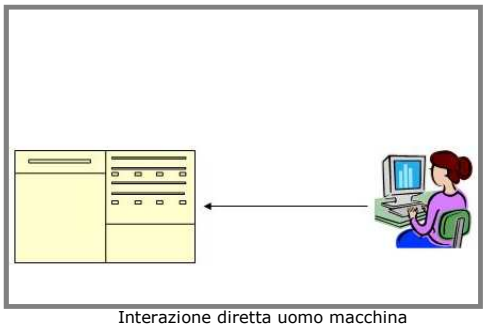
Il calcolatore e' un sistema per eseguire programmi

1. Caricare il programma in memoria
2. Memorizzare i dati
3. Eseguire il programma
4. Stampare (o visualizzare) i risultati



Fasi di esecuzione dei programmi

Necessita' di interagire direttamente con i dispositivi hardware



Interazione diretta uomo macchina

Ruolo dei sistemi operativi

Un sistema operativo e' un ambiente **software** che agisce da **intermediario tra l'utente e il computer**

Dal punto di vista del sistema:

Gestisce le risorse hardware in modo equo ed efficiente.

Dal punto di vista dell'utente:

Rende il sistema di calcolo semplice da usare senza fare riferimento alle specifiche fisiche del sistema

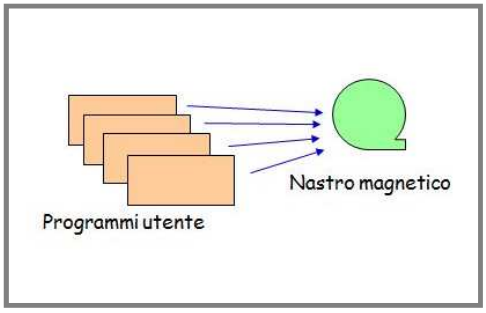
Come si sono sviluppati i sistemi operativi?

Ruolo del sistema operativo in un sistema di calcolo

Anni '50: Sistemi batch

Conseguenze della interazione diretta tra utente e sistema di calcolo

- Elaborazione lenta e inefficiente
- Un solo utente alla volta
- Alta possibilita' di errori
- Gestione inefficiente del sistema
- L'utente e' anche operatore del sistema



Preparazione di un lotto di programmi su nastro magnetico

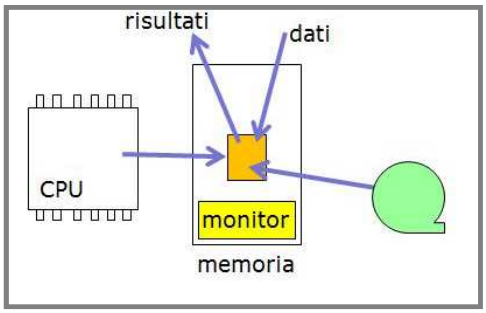
Per automatizzare il processo, i primi sistemi di calcolo memorizzavano su nastri magnetici gruppi di programmi (lotti o batch) che venivano eseguiti uno di seguito all'altro

Sistemi batch

Sistemi batch

Un programma sempre in esecuzione (**monitor**) gestiva la transizione da un programma ad un altro

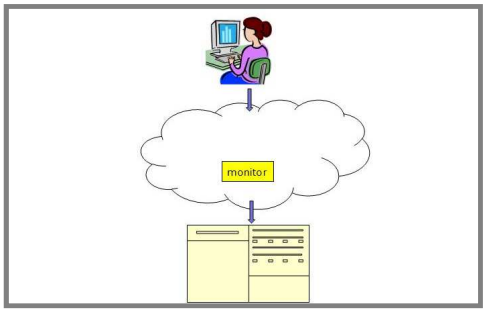
Il monitor e' il primo embrione di sistema operativo (primi anni '50)



Esecuzione batch sotto controllo del monitor

Con i sistemi batch si ottengono alcuni vantaggi:

- Utente diverso da amministratore
- Riduzione tempi di attesa tra utenti



Esecuzione batch sotto controllo del monitor

Anni '60: la multiprogrammazione

Problema:

Le fasi di I/O sono molto piu' lente delle fasi di calcolo, per cui la CPU era prevalentemente inattiva

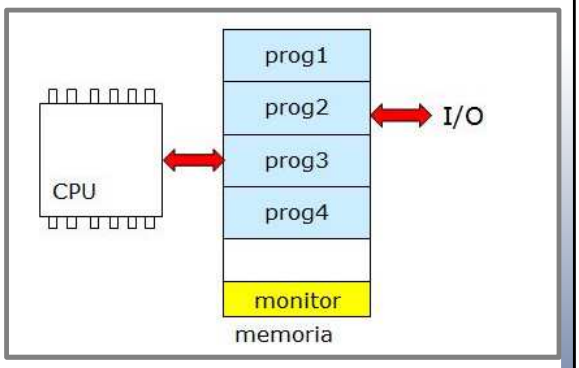
A partire dagli anni '60 si sono sviluppati i

Sistemi Operativi con Multiprogrammazione

(capacita' di tenere in memoria piu' programmi contemporaneamente)

Mentre un programma esegue operazioni di I/O (prog2), un altro programma (prog3) puo' utilizzare la CPU

Il **memory manager** e' incaricato di tenere traccia della memoria occupata



Organizzazione della memoria in un sistema con multiprogrammazione

Esempio di multiprogrammazione

3 programmi iniziano l'esecuzione nello stesso istante.

caratteristiche

- P1 richiede 24 msec di CPU
- P2 richiede 3 msec di CPU
- P3 richiede 3 msec di CPU

Tempi di completamento

- P1 termina dopo 24 msec
- P2 termina dopo 27 msec (24 attesa + 3 esecuzione)
- P3 termina dopo 30 msec (27 attesa + 3 esecuzione)

Tempo medio di completamento =

(24+27+30)/3 = 27 msec

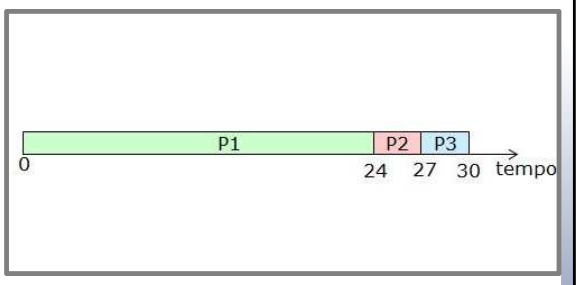


Diagramma dell'esecuzione di 3 processi in un sistema multiprogrammato

Il time sharing

Se in memoria e' presente un programma che richiede molto tempo di calcolo, altri programmi piu' brevi devono attendere la fine del primo programma (tempo medio di elaborazione elevato)

Soluzione

Assegnare ad ogni programma un
tempo massimo di utilizzo della CPU (time slice)

Se si supera tale limite il programma viene sospeso
e si manda in esecuzione un altro programma
(time sharing)

Un programma del sistema operativo (**dispatcher**)
si occupa della sospensione e della ripresa dei programmi

Esempio di time sharing

3 programmi iniziano l'esecuzione nello stesso istante. Time slice = 5 msec

caratteristiche

- P1 richiede 24 msec di CPU
- P2 richiede 3 msec di CPU
- P2 richiede 3 msec di CPU

Tempi di completamento

- P1 termina dopo 30 msec (6 attesa + 24 esecuzione)
- P2 termina dopo 8 msec (5 attesa + 3 esecuzione)
- P3 termina dopo 11 msec (8 attesa + 3 esecuzione)

Tempo medio di completamento =
(30 + 8 + 11)/3 = 16,33 msec

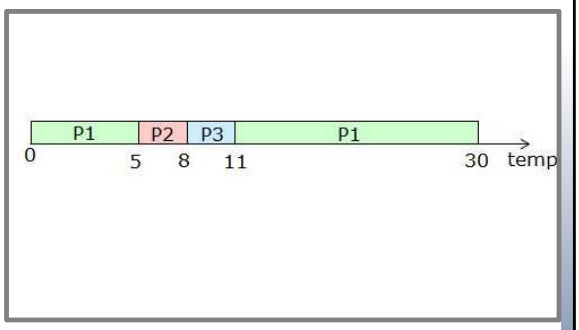


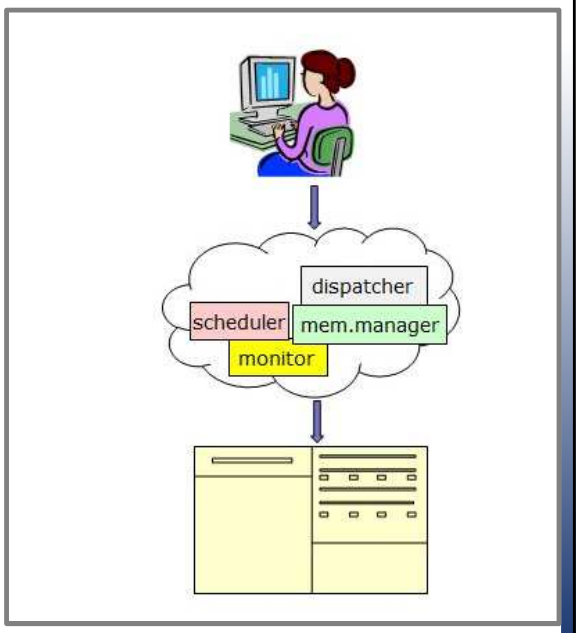
Diagramma dell'esecuzione di 3 processi in un sistema multiprogrammato e time sharing

Multiprogrammazione e time sharing

Un programma (**scheduler**)
si occupa dell'ordine con cui i programmi devono essere in esecuzione

Con la multiprogrammazione
e il time sharing
si riducono i tempi medi di attesa
e ogni utente ha la sensazione di essere l'unico utente del sistema

Il primo sistema operativo completo con multiprogrammazione e time sharing e' stato il Multics (1964)



Un sistema operativo
multiprogrammato con time sharing

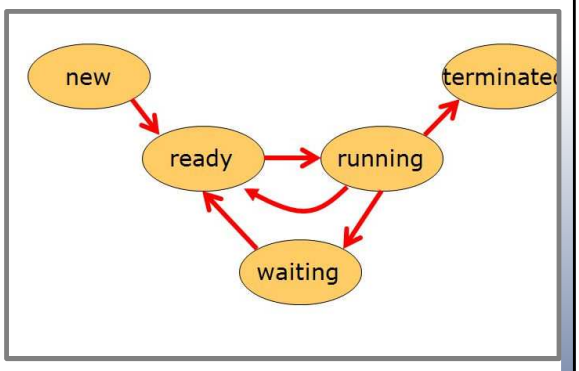
Ciclo di vita dei processi

Un programma in esecuzione prende il nome di **processo**

Un processo attraversa varie fasi:

- **New** (viene allocato lo spazio in memoria)
- **Ready** (il processo e' pronto, ed aspetta il suo turno)
- **Running** (il processo utilizza la CPU)
- **Waiting** (il processo attende il completamento di una operazione di I/O)
- **Terminated** (viene liberato lo spazio in memoria)

sempre un solo processo running
molti processi ready o waiting



Ciclo di vita dei processi

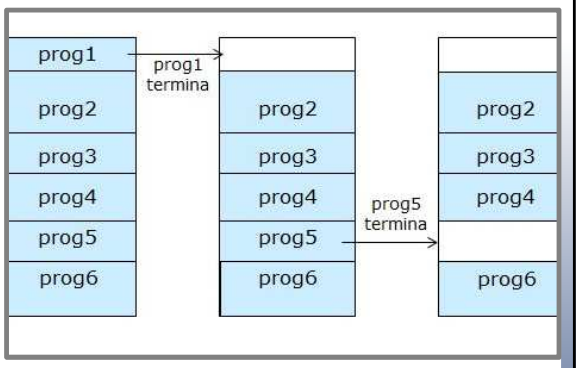
Frammentazione della memoria

Con la multiprogrammazione si alloca ad un processo uno spazio della dimensione del processo stesso

Quando i processi finiscono si creano dei buchi (holes) in memoria

Parecchio spazio disponibile in memoria ma non utilizzabile perche' non contiguo

FRAMMENTAZIONE



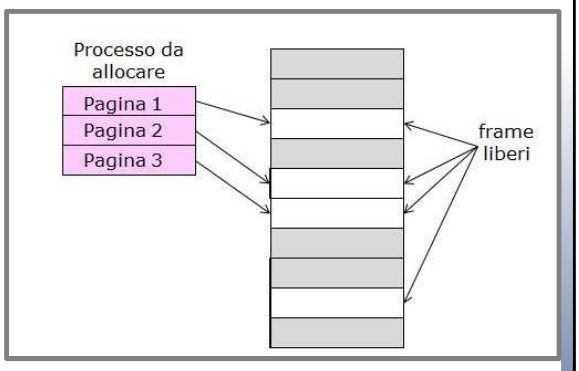
Esempio di frammentazione della memoria

La memoria virtuale

Allocazione non contigua

- permette di allocare la memoria fisica ai processi ovunque essa sia disponibile.
- Si divide la memoria fisica in blocchi di dimensione fissa chiamati blocchi (o frame)
- Si divide il processo in blocchi della stessa dimensione chiamati pagine
- Per eseguire un processo di n pagine, è necessario trovare n frame liberi prima di caricare il programma.

MEMORIA VIRTUALE



Allocazione di un processo in memoria con la tecnica della memoria virtuale

Problema:

Il time sharing e la memoria virtuale permettono l'accesso alla CPU a molti programmi residenti in memoria

Ma la memoria ha una capacita' limitata

Come fare a tenere numerosi programmi (anche grandi) contemporaneamente in memoria?

conservare:

- in memoria centrale solo la pagine con la sezione di codice da eseguire
- in memoria di massa il resto delle pagine

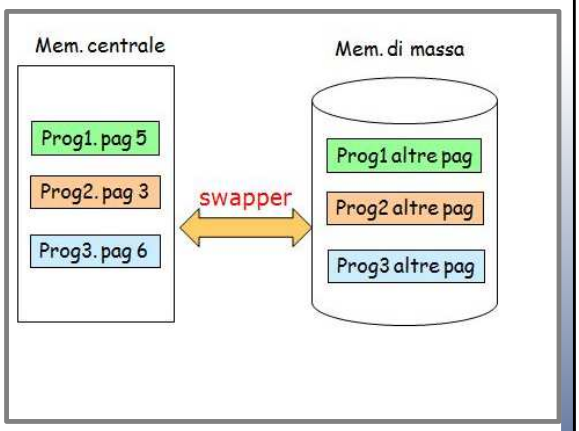
Un programma del sistema operativo (**swapper**) si occupa della sostituzione delle pagine in memoria

Ruolo dello swapper

La memoria virtuale e' lo spazio di indirizzamento effettivamente utilizzabile dall'utente

- Automazione dello scambio tra memoria centrale e disco
- Lo spazio logico degli indirizzi e' più grande dello spazio fisico
- Un maggior numero di processi puo' concorrere all'uso della CPU

- Meccanismi analoghi esistono per lo scambio di informazioni tra memoria centrale e memorie cache



Schema di memoria virtuale

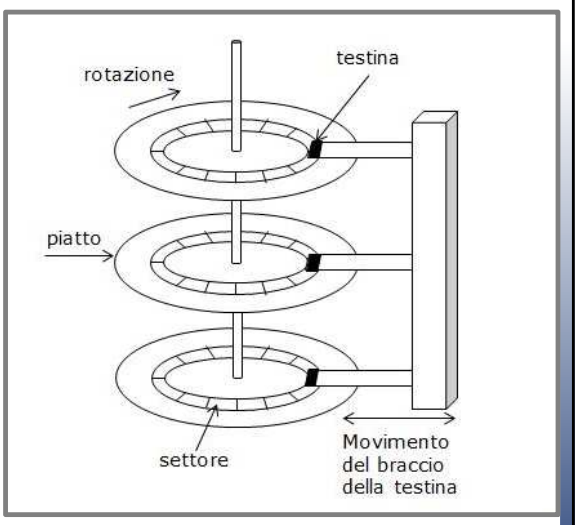
L'organizzazione del disco

Oltre alla memoria centrale i calcolatori necessitano di un supporto di di piu' grandi dimensioni e non volatile

La memoria secondaria
(disco rigido – hard disk)

I dischi sono costituiti da piu' piatti, organizzati come sequenza di settori logici, che rappresentano le minime unità di trasferimento (dim. tipica 512 byte)

La rotazione del disco e lo spostamento di una testina permettono l'accesso ai dati memorizzati nei settori



Uno schema di disco rigido

Il file system

Dal punto di vista dell'utente, le informazioni sono organizzate in strutture chiamati file, che possono essere distribuiti su piu' settori del disco

il file e' la piu' piccola unita' di memoria secondaria accessibile agli utenti
(tutti i dati presenti nella memoria secondaria devono essere organizzati in file)

Un sistema operativo puo' gestire

- milioni di file,
- di differenti utenti
- di tipi differente

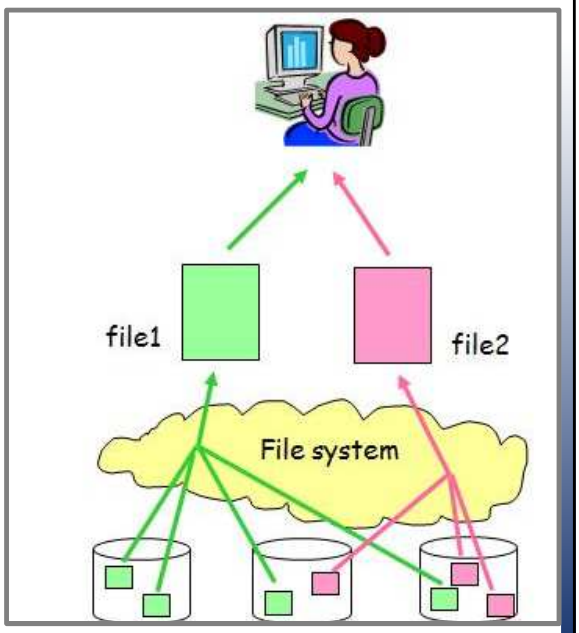
Necessita' di organizzare logicamente i file

L'organizzazione logica dei file di un sistema operativo e' chiamata
FILE SYSTEM

Il file system

- Organizza i file e gestisce l'accesso ai dati
- Responsabile della integrita' dei file, della gestione dei metodi di accesso (lettura/scrittura)
- Nasconde all'utente l'organizzazione fisica del disco

Visione dei file indipendente dai dispositivi fisici !!!



Modello di file system

Struttura logica del file system

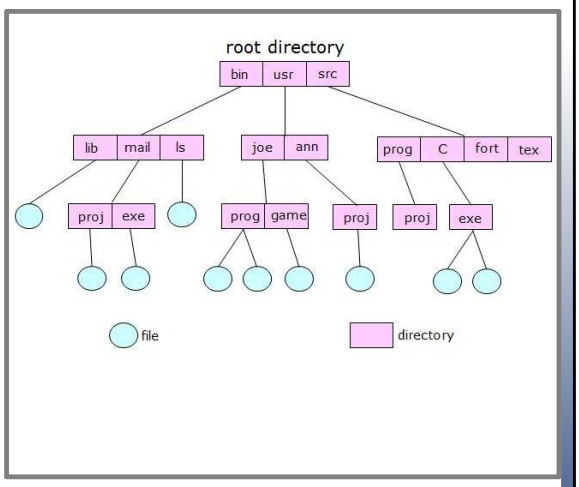
Lo strumento piu' comune usato per realizzare un file system e' la directory (elenco o cartella)

Di solito, gli elementi di una directory possono essere file o altre directory in maniera da creare una

struttura ad albero

Una directory principale (root directory, master file directory) indica la radice dell'albero dalla quale partono le directory di secondo livello

Windows e Linux hanno tale organizzazione



Un file system organizzato ad albero

Dispositivi di I/O

Un sistema di calcolo si interfaccia con il mondo esterno attraverso i **dispositivi di I/O**

(monitor, tastiera, mouse, CD/ROM, stampanti, scanner, pen drive USB,…)

Grande varieta' di dispositivi basati su pochissime operazioni

- **Input**: Leggi dati da un supporto esterno
- **Output**: scrivi dati su un supporto esterno

Alcuni compiti del sottosistema di I/O

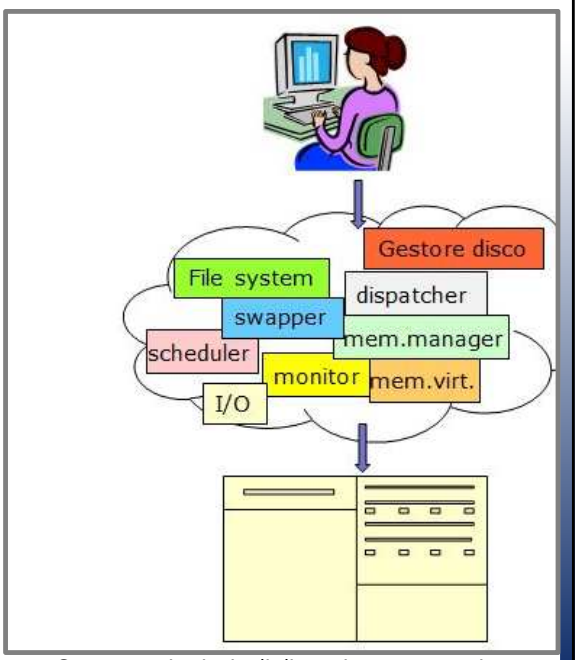
- Interfacciamento uniforme (far apparire i dispositivi simili tra loro)
- Bufferizzazione (contribuire a gestire le differenti velocita' dei dispositivi)
- Gestire e riportare errori (es. lettura da unita' di output e viceversa)
- Indirizzamento delle varie unita' di I/O

Componenti di un sistema operativo moderno

La struttura dei moderni sistemi operativi e' quindi il frutto di aggiunte di programmi introdotti nel tempo per far fronte a esigenze di efficienza (per il sistema) e semplicita' di uso (per l'utente)

Principali sottosistemi di un sistema operativo

- Gestione dei programmi in esecuzione
- Gestione della memoria centrale
- Gestione della memoria di massa e del file system
- Gestione dell'I/O



Componenti principali di un sistema operativo

Interprete del linguaggio di comando

Il sistema operativo gestisce il sistema per conto dell'utente

Sistema operativo e utente comunicano attraverso il

Linguaggio di Comando

Ogni comando e' una richiesta al sistema operativo per eseguire una certa operazione

Un **interprete del linguaggio di comando** si occupa dell'interpretazione del comando e del compito da seguire

L'interprete del linguaggio di comando

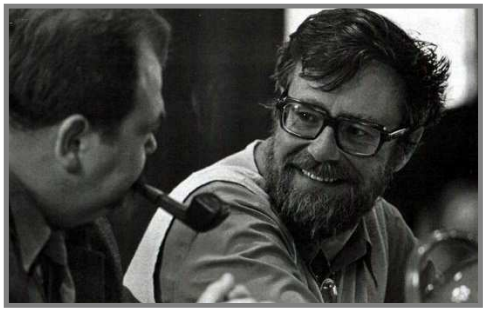
riassumendo

- Il sistema operativo e' l'intermediario software tra utente e sistema di calcolo
- Obiettivi:
  - Gestione efficiente delle componenti hardware
  - Semplicita' di utilizzo per l'utente
- Principali compiti
  - Gestione dei processi (gestione della CPU)
  - Gestione della memoria
  - Gestione della memoria di massa e del file system
  - Gestione dell'I/O

Un po' di storia (6)

Edsger Dijkstra (1930-2002)

- Laureato in fisica in Olanda, e' professore universitario di informatica ad Eindhoven.
- Negli anni '60 , sviluppa il concetto di semaforo per la sincronizzazione dei processi nei sistemi operativi, e formulo' il problema dei "Filosofi a cena"
- Con il suo articolo "Go To Statement Considered Harmful" pone le basi della programmazione strutturata , che influenzera' le successive generazioni dei linguaggi di programmazione
- Fu anche il promotore della verifica formale degli algoritmi, un metodo che consiste nel sviluppare contemporaneamente l'algoritmo e la dimostrazione della sua correttezza
- Vince il premio Turing nel 1972 per i suoi studi sulla struttura dei linguaggi di programmazione



E. Dijkstra nel 1972 (dijkstracry.com)

Nel 1994 durante una lezione (Wikimedia Commons)