Corso Sistemi Operativi AA2017-2018

Prof. Enzo Mumolo

Motivazioni principali del corso

- Il corso NON si occupa di utilizzo dei Sistemi Operativi a livello utente

MA

- Di conoscenza approfondita del linguaggio C
- Di conoscenza dell'interprete di comandi Bash
- Di familiarizzare con la Programmazione di sistema in Linux
- Di conoscenza approfondita delle Chiamate di Sistema di Linux

Concetto di Processo: istanza di un programma in esecuzione

- Processi concorrenti non possono interferire
- Esempio: due programmi indipendenti: fattoriale e determinare se un intero è primo

```
int Fatt(int n) {
          int k=1;
          for (i=1; i <= n; i++)
             k = k* i;
          return(f);
}
```

```
static boolean isPrime(int number) {
    if (number == 1)return false;
    if (number == 2)return true;
    if (number % 2 == 0)return false;
    for (int d=3; d<=(int)Math.sqrt(number); d++){
            if (number % d == 0)return false;
    }
    return true;
}
```

- Sequenze di istruzioni del primo:
  - per input 5: [k=1] [k=1*1] [k=1*2] [k=2*3] [k= 6*4] [k=24*5]
- Sequenze di istruzioni (diverse per ogni input):
  - per input 17: [if(17==1)] [if(17==2)] [if(17%2)] [d=3] [if(17%3)] [d=4] [if(17%4)] [true]
  - per input 13: [if(13==1)] [if(13==2)] [if(13%2)] [d=3] [if(13%3)] [true]
- Programmazione concorrente: esecuzione concorrente con 2 core dei due processi:
  - [k=1] [k=1*1] [k=1*2] [k=2*3] [k= 6*4] [k=24*5]
  - [if(13==1)] [if(13==2)] [if(13%2)] [d=3] [if(13%3)] [true]

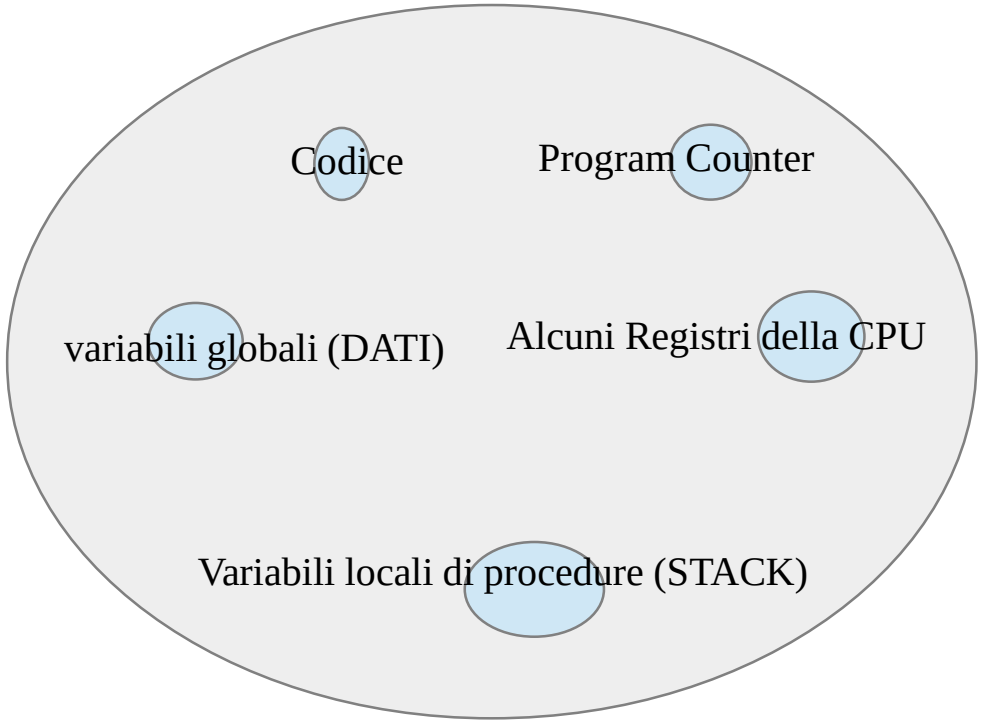
Concetto di processo

- Programmazione concorrente: esecuzione concorrente con 2 core dei due processi:
    - [k=1] [k=1*1] [k=1*2] [k=2*3] [k= 6*4] [k=24*5]
    - [if(13==1)] [if(13==2)] [if(13%2)] [d=3] [if(13%3)] [true]
- Quasi concorrenza (con 1 core):

```
[k=1] [k=1*1]                          [k=1*2] [k=2*3]                  [k= 6*4] [k=24*5]
            [if(13==1)] [if(13==2)]                  [if(13%2)] [d=3]
```

- Ad un processo sono associati:
    - Uno o più segmenti di *'codice eseguibile'*.
    - Uno o più segmenti di memoria dati.
    - I descrittori di eventuali risorse in uso (file, finestre, periferiche, ecc.)
    - Uno o più thread.
- Queste informazioni sono contenute nel PCB (Process Control Block)

Componenti di un processo

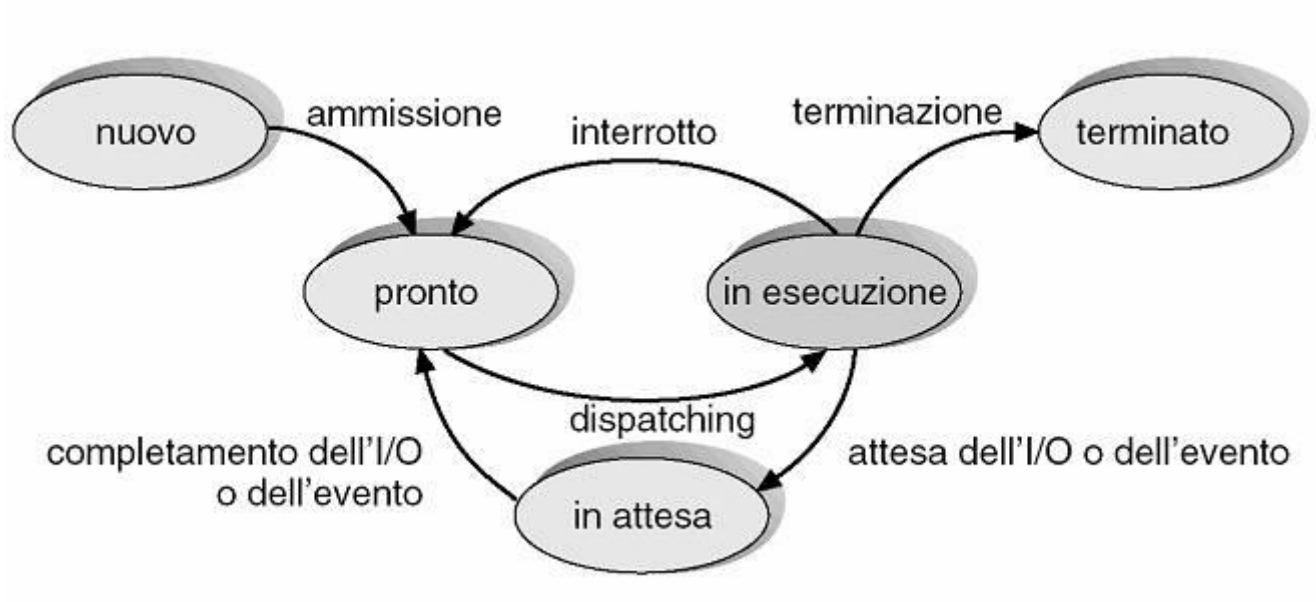


Processo

Stato del processo	Identificativo processo	Program counter	registri CPU	puntatori alla memoria	risorse aperte	tempi di esecuzione

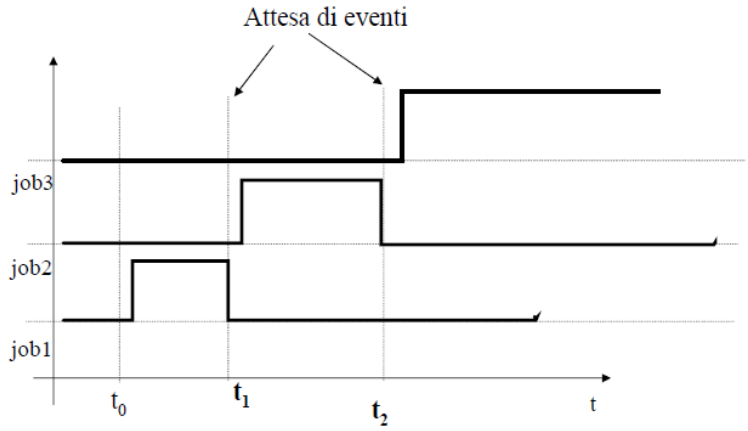
Descrittore di processo: Process Control Block PCB

Stati di un processo

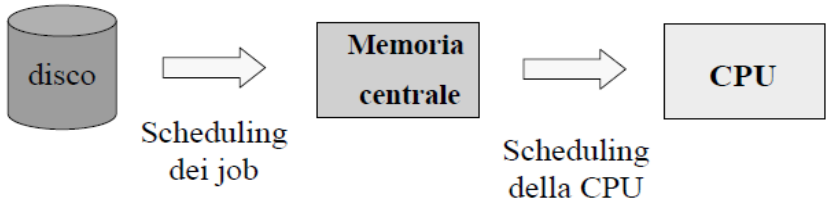


Elaborazione Batch

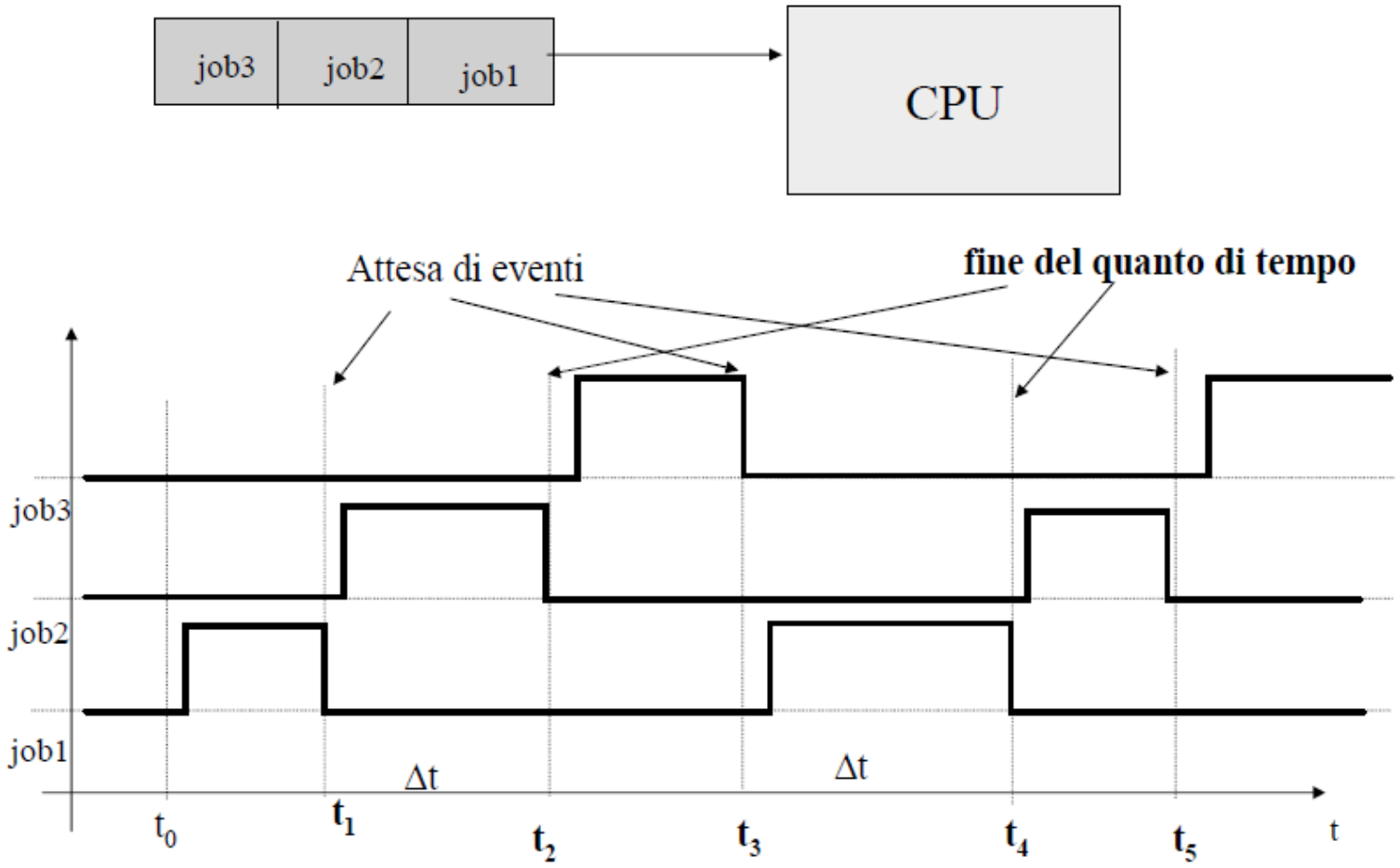
- Batch semplici
  - sistema operativo residente in memoria (monitor)
  - assenza di interazione tra utente e job
  - scarsa efficienza:durante l'I/O del job corrente, la CPU
- Batch multiprogrammati



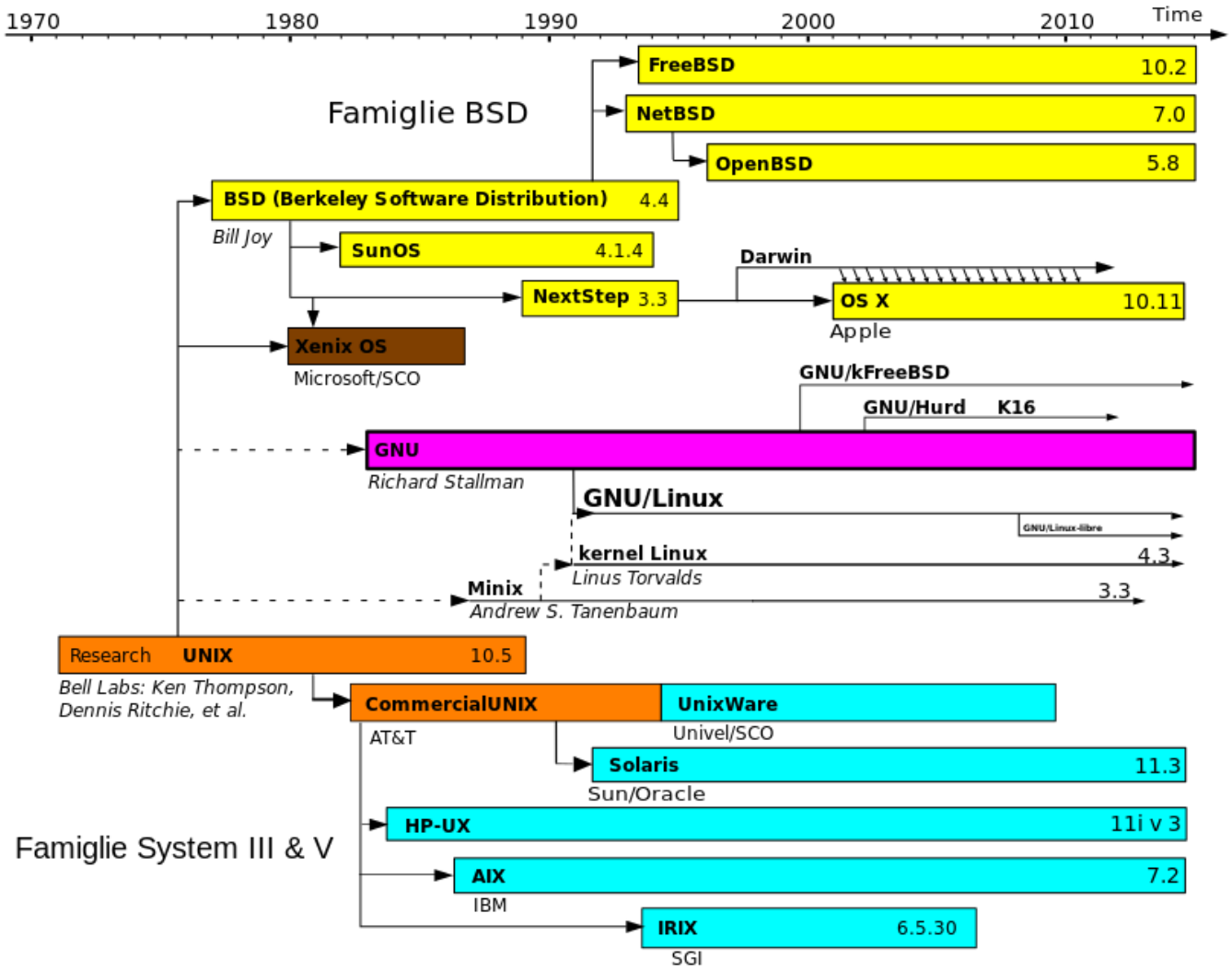
  - schedulazione



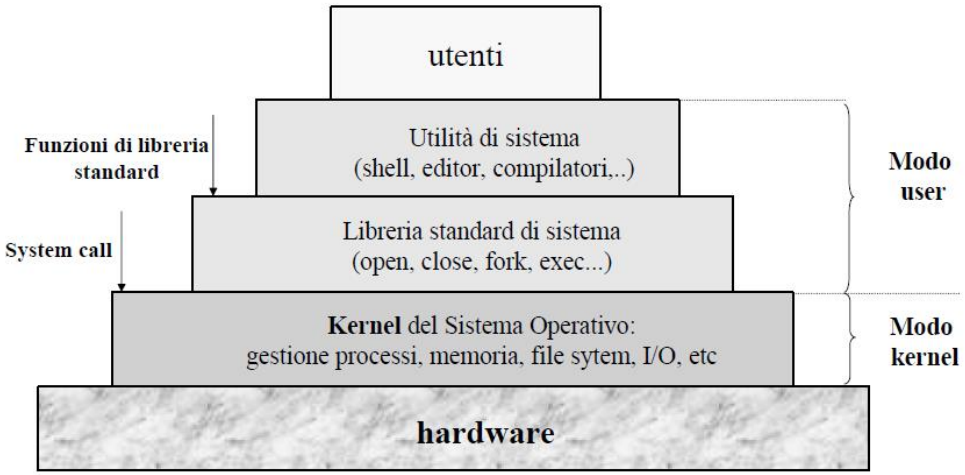
Elaborazione time-sharing



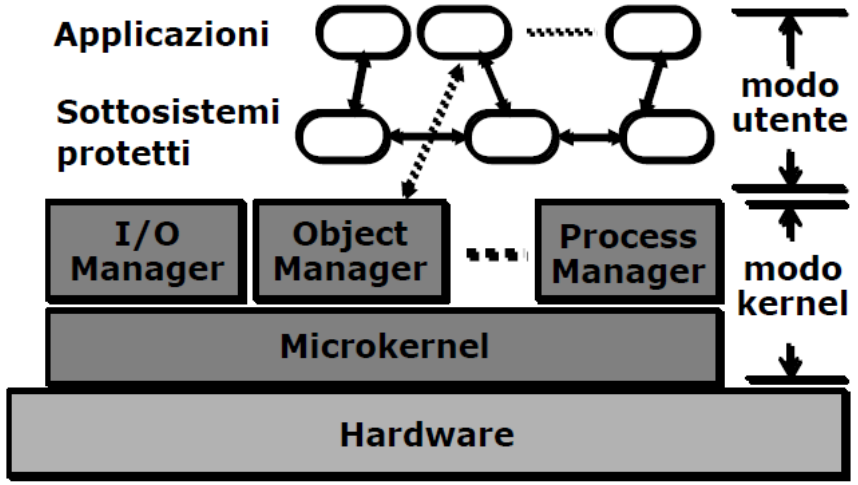
Evoluzione Unix



Strutture di Unix e Windows - schemi

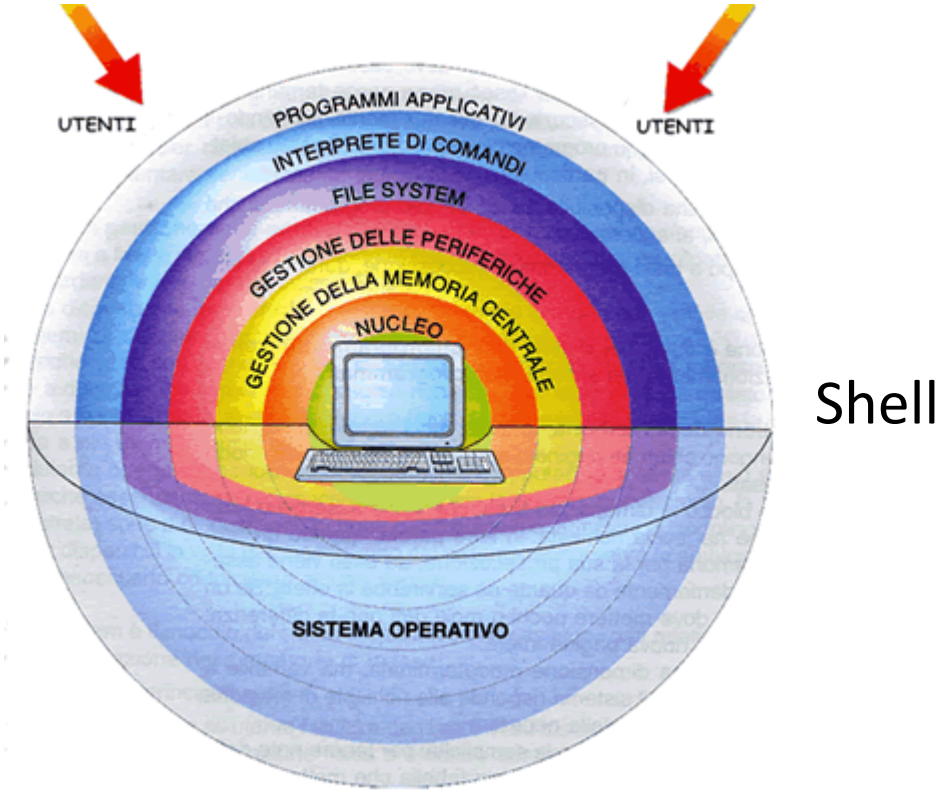


Unix



Windows NT

Struttura di THE, VAX/VMS



Shell

Shell

Struttura Onion-Skin

Breve cronistoria dei Sistemi Operativi

- 1964 OS/360: paginazione, batch, spooling, multiprog, milioni di righe ASM
- 1965 Multics: time-sharing, schedulazione, segmentazione
- 1968 THE (*Technische Hogeschool Eindhoven):* struttura Onion Skin, stallo
- 1969 Unix : monolitico, C, tre livelli (User,System,hw), time-sharing
- 1977 BSD (Berkeley Software Distribution): serie di patch a Unix per PDP-11
- 1978 VAX/VMS (Virtual Memory System): struttura Onion-Skin
- 1979 Apple DOS 3.2: usato negli Apple II
- 1981 MS-DOS: Disk OS, single-user, single-tasking, 640KB utente, 360 k sistema

  Microsoft → IBM

Breve cronistoria dei Sistemi Operativi (cont'd)

- 1982 SUN OS
- 1984 MacIntosh OS
- 1985 Microsoft Windows 1
- 1987 Minix, OS2 1.0 (IBM), Windows 2
- 1990 Windows 3
- 1991 Kernel Linux, OS2 1.3
- 1992 OS2 2.0, Windows 3, Solaris
- 1993 FreeBSD, NetBSD, Windows NT

Breve cronistoria dei Sistemi Operativi (cont'd)

- 1994 OS 2.11, RedHat Linux
- 1995 OS/390, Windows 95
- 1996 Debian GNU Linux
- 1997 Mac OS
- 1998 Windows 98
- 2000 Windows 2000, MacOS X
- 2001 Windows XP
- 2003 Windows Server 2003

Breve cronistoria dei Sistemi Operativi (cont'd)

- 2004 Ubuntu 4
- 2005 Gentoo Linux, SuSe, Ubuntu 5, Windows XP Pro
- 2006 Ubuntu 6, SuSe 10
- 2007 Ubuntu 7, Windows Vista
- 2009 Ubuntu 7, Windows 7
- 2012 Windows 8
- 2014 Ubuntu 14
- 2015 Windows 10, Ubuntu 15

Perché Linux?

- Il corso si basa sul S.O Linux
- Linux si basa su un nucleo (kernel) che è utilizzato in moltissimi Sistemi Operativi.
- I più popolari: Mac OS (2001-2007), OS X (2009-2015), macOS (2016-2018), FreeBSD, SuSe, OpenSuSe, Ubuntu e tutte le distribuzioni di Linux, Android …
- I kernel di Linux sono scaricabili gratuitamente dal sito ww.kernel.org

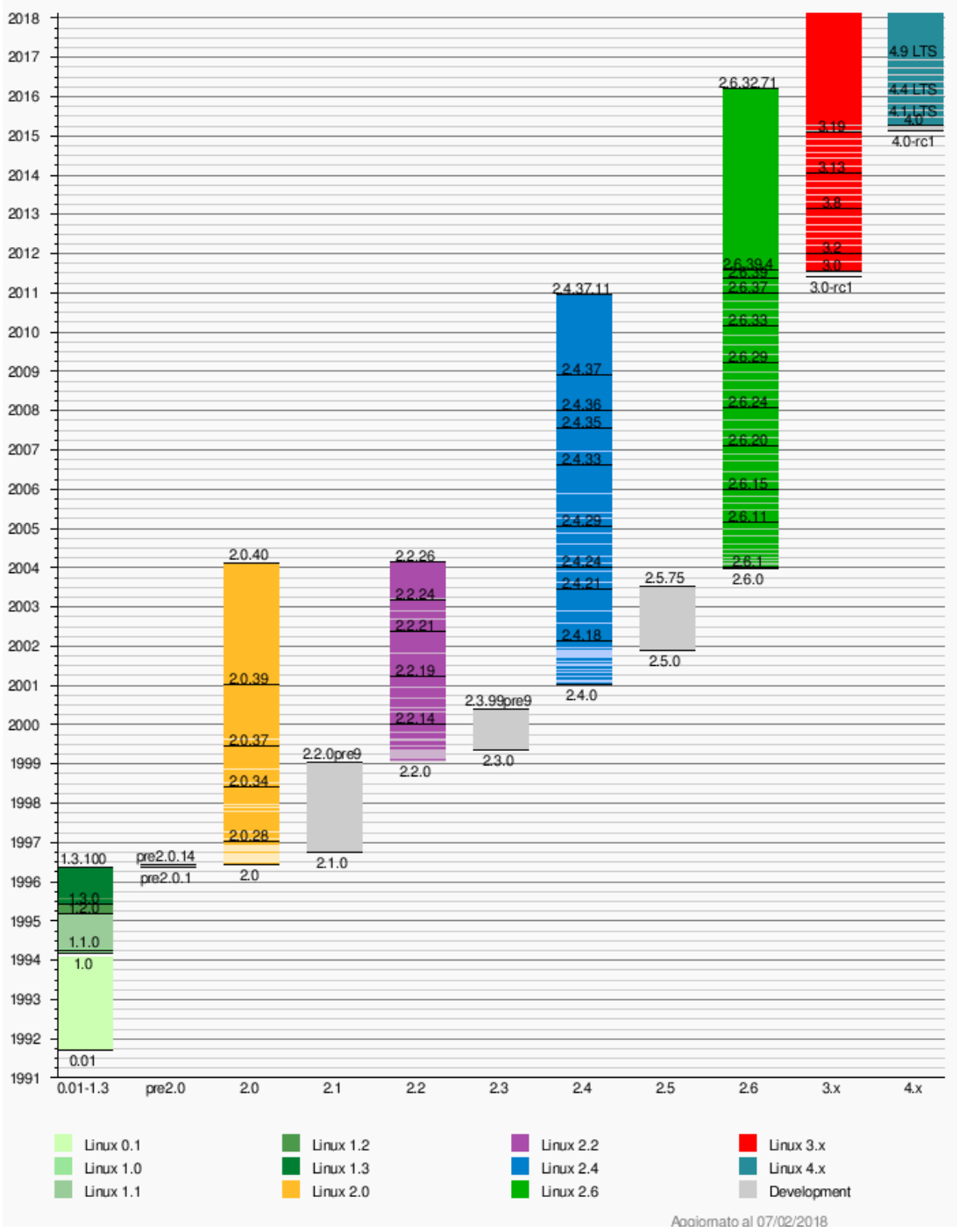
Perché Linux?

- Il kernel è stato scritto dal programmatore finlandese

  Linus Torvald (1970) partendo dal Minix
- Tesi di laurea magistrale (Helsinki-1991)

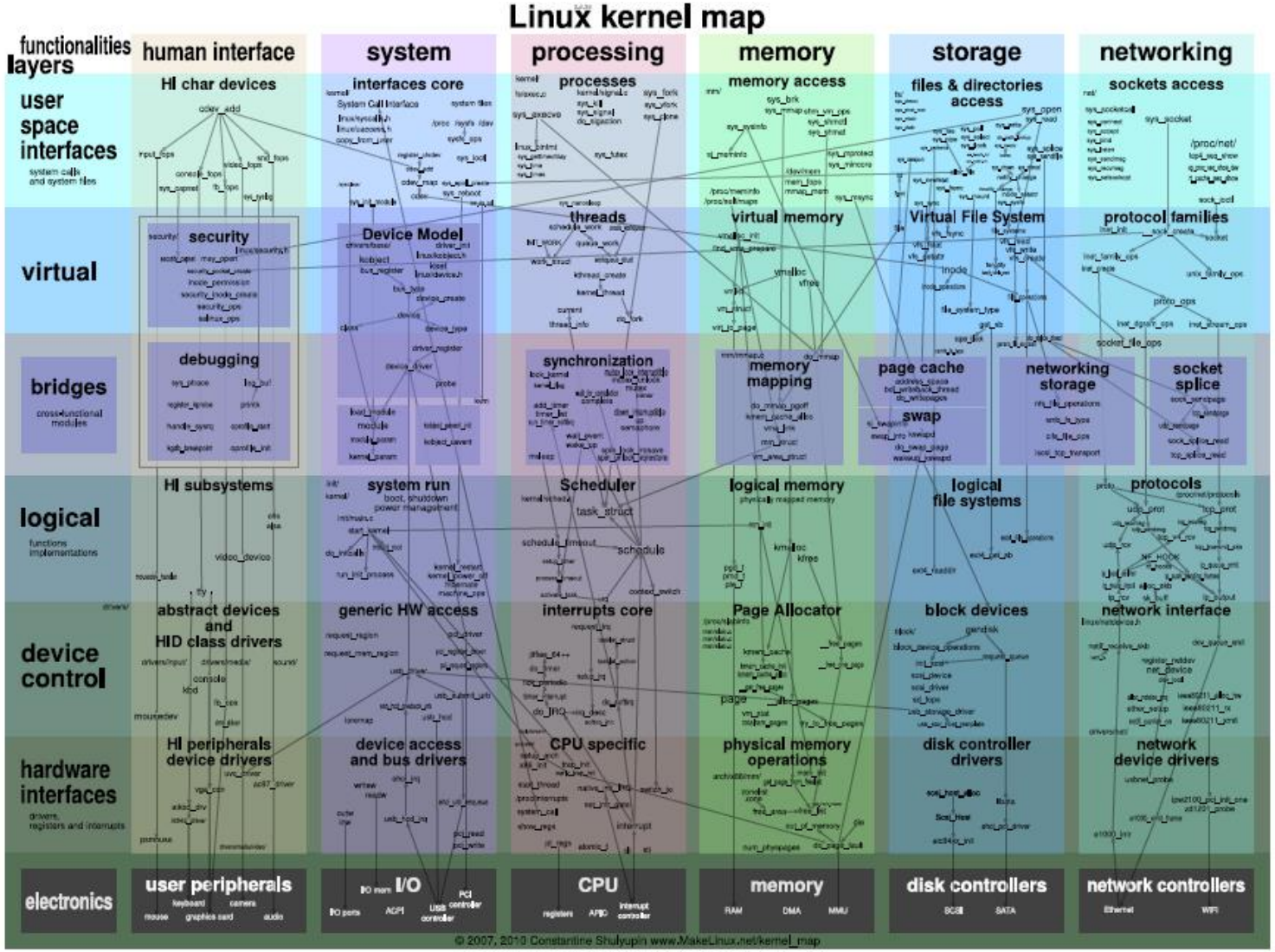
  ‘Linux: A Portable Operating System’
- Chiave di successo di Linux: il suo codice è pubblico, con licenza d'uso che consente a chiunque la libera modifica del codice.
- Torvald coordina gli aggiornamenti del codice del kernel usando il sito *www.kernel.org*

Linux Versioning



Una differenza tra Unix e Linux

- Linux è strutturato in moduli per manutenzione, aggiornamento, device driver



Perché Linux?

- I principi della programmazione di sistema condivisi da altri SO, ma con diversa complessità
- Per esempio, per gestire i processi:

	Linux	Windows
Creazione processi	fork(), exec()	CreateProcess, CreateProcessAsUser CreateProcessWithLogonW CreateProcessWithTokenW
Attesa e Cancellazione processi	wait(), exit()	TerminateProcess
Interprocess Communication	Pipe, Fifo, Shared memory, socket	Clipboard, Data Copy, Dynamic Data Exchange (DDE), Component Object Model COM) File Mapping, Mailslots, Pipes, Remote Procedure Call (RPC), Sockets

Perché Linux?

- Negli ultimi anni, si sono diffuse molte schedine basate sul processore ARM
- Queste schede hanno:
  - Piccole dimensioni
  - Basso consumo
  - Alte prestazioni
  - Basso costo
  - Schema interno pubblicato su internet
  - Linux come sistema operativo (perché gratuito)
- Vengono utilizzate o realizzate da molte aziende

Programma del corso (6CFU)

- Informazioni sull'esame e concetti preliminary
- Cenni di architetture dei calcolatori
- Richiami del linguaggio C
- Aritmetica dei puntatori
- Confronto tra linguaggio C e linguaggio Java
- Traduzioni di programmi tra i due linguaggi
- Struttura del file system di Linux file system di Linux
- Identificazione utente, file, processi
- Comandi fondamentali di Linux
- Programmazione in Bash
- Chiamate di sistema per la gestione dei file in Linux
- Chiamate di sistema per la gestione dei processi in Linux
- Chiamate di sistema per la gestione delle Inter Process Communication
- Syncronizzazione tra processi in Linux
- Problemi classici di sincronizzazione
- Realizzazione di Shell in C
- Intropuzione ai Pthread

Modalità di valutazione dell’esame

- Valutazione di 6 CFU
  - Due provette durante l’anno
  - Se entrambe le provette sono positive → media
  - Possibilita' di aumentare il voto con tesina aggiuntiva
  - Se una tesina è positiva e l'altra è negativa da 14/30 a 18/30 → orale sugli esercizi negativi
  - Se una tesina è positiva e l'altra è negativa < 14/30 → orale o scritto su tutto il programma della provetta
  - Se entrambe sono negative < 18/30 → altra modalità d'esame:
    - Scritto/orale su tutto il programma
    - Scritto/orale su tutto il programma
    - Tesina/orale su tutto il programmma
- Gli eventuali aggiuntivi 3 CFU vengono vautati con tesine
- Il voto finale si ottiene come media del voto dei 3CFU e del voto dei 3 CFU