





Guanciale 150 gr.  
Olio 2 cucchiari  
Pecorino 100 gr.  
Pepe secondo i gusti  
Spaghetti 350 gr.  
Uova 4 tuorli + 1 intero

La ricetta della pasta alla carbonara prevede l'utilizzo di determinati ingredienti e lo svolgimento di precise procedure. Tuttavia eventuali varianti possono arricchire il gusto modificandolo a seconda delle preferenze dei commensali

Se la carbonara avesse un copyright tutte le varianti dovrebbero essere autorizzate dal proprietario dei diritti. Copiare o modificare la ricetta della pasta alla carbonara senza autorizzazione violerebbe i termini di utilizzo e sarebbe illegale.

# 1960-1970

- E' stato scritto il linguaggio di programmazione C
- Sono state poste le basi della attuale Internet
- Vengono introdotti sul mercato i primi minicomputer
- Nei laboratori Bell Labs viene sviluppato UNIX
- Si crea una comunità di ricercatori che sviluppano software per i calcolatori delle università connesse da Arpanet.

Le procedure ed il software stesso venivano scambiati liberamente.

Gli sviluppatori venivano pagati per il lavoro di programmazione e non per il software vero e proprio.

# 1970



Nel 1969, nei laboratori della Bell Labs, una divisione della AT&T, viene sviluppato UNIX

L'Antitrust americano vieta all AT&T di inserirsi nel settore informatico  
UNIX viene distribuito a prezzo simbolico nelle università americane, riscritto in C  
e compreso di codice sorgente

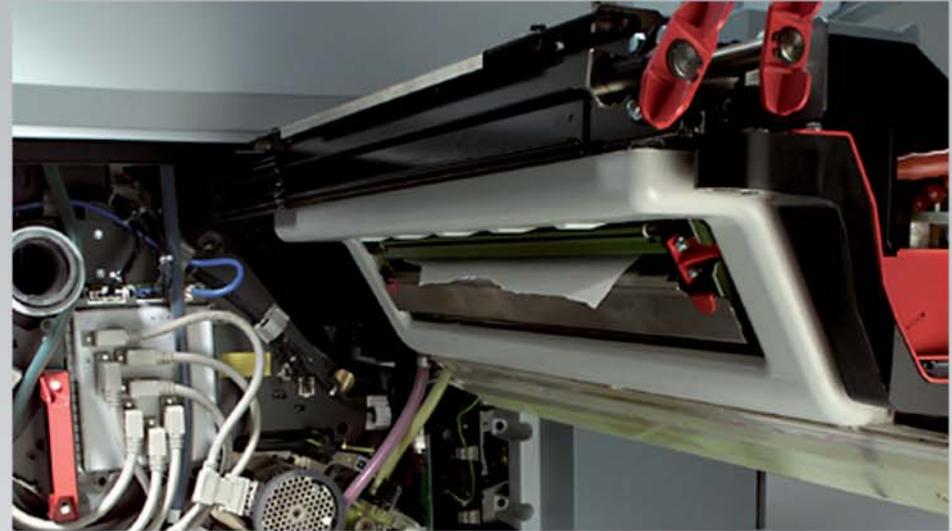
Dal 1978 L'università di Berkley rilascia la sua versione di UNIX la BSD  
(Berkley Software Distribution)

AT&T suddivide la società in 26 controllate e aggira l'Antitrust

La distribuzione di UNIX acquisisce delle caratteristiche del tutto commerciali  
vengono aumentati i costi di licenza e omessa la possibilità di modifica

Singoli produttori di hardware vendono le loro macchine con una versione di UNIX  
leggermente modificata e senza codice sorgente





Nel 1980 il laboratorio di Intelligenza Artificiale al MIT rinnovò la dotazione di stampanti.

Le nuove Xerox 9700 avevano precluso l'accesso al codice sorgente dei driver.

A Richard Stallman fu impedito di ripetere l'operazione di modifica dei driver per la segnalazione della carta inceppata e del termine lavoro.

# 1980

La ditta Symbolics chiese di poter utilizzare l'interprete per il linguaggio Lisp a Stallman, il quale accettò fornendo loro una versione pubblica della sua opera.

Symbolics estese e migliorò il lavoro, ma quando Stallman volle accedere ai miglioramenti Symbolics rifiutò. Inizia la battaglia per sradicare la pratica che Stallman definisce di “accaparramento del software”

Nel 1984 viene lanciato il progetto **GNU (Gnu is not UNIX)**

L'idea è di sviluppare un sistema operativo Unix-compatibile completo che fosse software libero



I sistemi operativi tipo Unix sono costituiti da un insieme di applicazioni, librerie e strumenti di sviluppo, oltre a un programma utilizzato per allocare le risorse e comunicare con l'hardware, noto come kernel.

Hurd, il kernel di GNU, è in sviluppo, ma è ancora piuttosto lontano dall'essere pronto per l'uso quotidiano, quindi GNU è spesso utilizzato con un altro kernel, di nome Linux.

La combinazione di GNU e Linux è il sistema operativo GNU/Linux, ora usato da milioni di persone e talvolta erroneamente chiamato solo “Linux”.

Fulcro di tutta l'attività del Progetto GNU è la licenza chiamata **GNU General Public License (GNU GPL)**, che sancisce e protegge le libertà fondamentali che, secondo Stallman, permettono l'uso e lo sviluppo collettivo e naturale del software.

# 1980



# FREE SOFTWARE FOUNDATION

Nel 1985 Stallman fonda la Free Software Foundation e delinea le caratteristiche che rendono un software “libero”

L’espressione “Software Libero” si riferisce alla libertà dell’utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software. Più precisamente, esso si riferisce a quattro tipi di libertà per gli utenti del software:

- Libertà di eseguire il programma, per qualsiasi scopo (libertà 0).
- Libertà di studiare come funziona il programma e adattarlo alle proprie necessità (libertà 1). L’accesso al codice sorgente ne è un prerequisito.
- Libertà di ridistribuire copie in modo da aiutare il prossimo (libertà 2).
- Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio (libertà 3). L’accesso al codice sorgente ne è un prerequisito.

Il “Software Libero” è una questione di libertà, non di prezzo. Per capire il concetto, bisognerebbe pensare alla “libertà di parola”.

# 1990



Message from discussion [What would you like to see most in minix?](#)

**Linus Benedict Torvalds** [View profile](#)

[More options](#) Aug 26 1991, 8:12 am

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torva...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

I still maintain the point that designing a monolithic kernel in 1991 is a fundamental error. Be thankful you are not my student. You would not get a high grade for such a design :-)

Nel 1991 l'insoddisfazione riguardante alcuni applicativi di Minix, un sistema Unix su una piattaforma PC, il desiderio di approfondire le proprie conoscenze del processore Intel 386, scelto in quanto di minor costo e di maggiore diffusione rispetto alle piattaforme hardware per le quali erano disponibili sistemi operativi Unix, e l'entusiasmo per le caratteristiche tecniche di Unix stimolarono Linus Torvalds, studente al secondo anno di informatica presso l'Università di Helsinki, a sviluppare un proprio sistema operativo.

# 1990

Linux è un sistema operativo, libero di tipo Unix (o unix-like) costituito dall'integrazione del kernel Linux con elementi del sistema GNU e di altro software sviluppato e distribuito con licenza GNU GPL o con altre licenze libere. Il suo sviluppo è sostenuto da un'associazione senza fini di lucro, ovvero la Linux Foundation.

Linux è il nome del kernel sviluppato da Linus Torvalds e da collaboratori di tutto il mondo a partire dal 1991.

Integrato con i componenti già realizzati dal progetto GNU è stato utilizzato come base per la realizzazione dei sistemi operativi open source e delle distribuzioni che vengono normalmente identificate con lo stesso nome.

La scelta di integrare il primo kernel Linux al progetto GNU di Richard Stallman derivò dalla insoddisfazione di Torvalds per Minix, il sistema operativo Unix-like sviluppato da Andrew Tanenbaum, docente di Sistemi di rete ad Amsterdam.

Linus decise di rendere il sistema indipendente da Minix, anche perché non ne gradiva la licenza che lo rendeva liberamente utilizzabile solo a fini didattici.

Per integrare Linux al progetto GNU, Torvalds doveva cambiare licenza e adottare la GPL, fu così possibile ridistribuire il software e soprattutto avere il supporto di migliaia di sviluppatori.

Con l'evoluzione di ambienti desktop come KDE e GNOME, il sistema offre una interfaccia grafica simile a quella di Microsoft Windows o di Mac OS X, più vicina alle esigenze degli utenti meno esperti.

# 1990



Per incoraggiare l'uso di software con licenza libera nel mondo degli affari, Bruce Perens, Eric S. Raymond, Ockman e altri cominciarono nel 1997 a pensare di creare un gruppo autonomo a favore di una ridefinizione ideologica del software libero.

Fu coniato il termine "Open Source" al fine di evitare l'equivoco dovuto al doppio significato di free nella lingua inglese, visto che spesso veniva interpretato come "gratuito" invece che come "libero".

Venne accettata l'idea che l'Open Source fosse una metodologia di produzione software efficace.

Vengono definite da Raymond due modelli di sviluppo software: Cattedrale e Bazaar

Nel modello a Cattedrale il programma viene realizzato da un numero limitato di "esperti" che provvedono a scrivere il codice in quasi totale isolamento. Il progetto ha una suddivisione gerarchica molto stretta e ogni sviluppatore si preoccupa della sua piccola parte di codice.

Nel modello a Bazaar il codice sorgente della revisione in sviluppo è disponibile liberamente, gli utenti possono interagire con gli sviluppatori e se ne hanno le capacità possono modificare e integrare il codice. Lo sviluppo è decentralizzato e non esiste una rigida suddivisione dei compiti.



LifeTrac

