

# L'evoluzione di Linux



Centro Ricerche Frascati

11 febbraio 2013

Marco Cesati

Università di Roma Tor Vergata





## Il movimento del software libero

- Nel 1983 Richard Stallman avvia il **Progetto GNU** (GNU's **N**ot **U**nix)
  - Scrivere un intero SO “libero” da diritti d'autore e licenze
  - Compatibile con il SO Unix
  - Quasi completato nei primi anni '90: manca solo il nucleo del SO



## Il movimento del software libero

- Nel 1983 Richard Stallman avvia il **Progetto GNU** (GNU's Not Unix)
  - Scrivere un intero SO “libero” da diritti d'autore e licenze
  - Compatibile con il SO Unix
  - Quasi completato nei primi anni '90: manca solo il nucleo del SO
- Nel 1985 Stallman fonda la **Free Software Foundation (FSF)**
  - Promuovere la scrittura e la diffusione del software libero



## Il movimento del software libero

- Nel 1983 Richard Stallman avvia il **Progetto GNU** (**GNU's Not Unix**)
  - Scrivere un intero SO “libero” da diritti d'autore e licenze
  - Compatibile con il SO Unix
  - Quasi completato nei primi anni '90: manca solo il nucleo del SO
- Nel 1985 Stallman fonda la **Free Software Foundation (FSF)**
  - Promuovere la scrittura e la diffusione del software libero
- Nel 1989 Stallman scrive la prima versione della licenza **GNU GPL** (**General Public License**)
  - La seconda versione è del 1991, la terza versione è del 2007
  - È la più diffusa licenza utilizzata per il **software libero**



## Il movimento del software libero

- Nel 1983 Richard Stallman avvia il **Progetto GNU** (GNU's Not Unix)
  - Scrivere un intero SO “libero” da diritti d'autore e licenze
  - Compatibile con il SO Unix
  - Quasi completato nei primi anni '90: manca solo il nucleo del SO
- Nel 1985 Stallman fonda la **Free Software Foundation (FSF)**
  - Promuovere la scrittura e la diffusione del software libero
- Nel 1989 Stallman scrive la prima versione della licenza **GNU GPL** (General Public License)
  - La seconda versione è del 1991, la terza versione è del 2007
  - È la più diffusa licenza utilizzata per il **software libero**



Le motivazioni di Stallman e dei proponenti del **software libero** sono primariamente di ordine “etico” e a salvaguardia dei diritti degli utenti

## Open Source Initiative (OSI)

- 1998: B. Perens e E.S. Raymond fondano la [Open Source Initiative](#)

## Open Source Initiative (OSI)

- 1998: B. Perens e E.S. Raymond fondano la [Open Source Initiative](#)
- Nasce per presentare al mondo industriale e commerciale una visione meno radicale ed idealistica di quella della [Free Software Foundation](#)



## Open Source Initiative (OSI)

- 1998: B. Perens e E.S. Raymond fondano la [Open Source Initiative](#)
- Nasce per presentare al mondo industriale e commerciale una visione meno radicale ed idealistica di quella della [Free Software Foundation](#)
- Lo scopo è comunque quello di promuovere software “aperto” che l'utente finale possa studiare ed adattare

## Open Source Initiative (OSI)

- 1998: B. Perens e E.S. Raymond fondano la [Open Source Initiative](#)
- Nasce per presentare al mondo industriale e commerciale una visione meno radicale ed idealistica di quella della [Free Software Foundation](#)
- Lo scopo è comunque quello di promuovere software “aperto” che l'utente finale possa studiare ed adattare
- La definizione di [software libero](#) e la definizione di [software open source](#) non coincidono esattamente
  - Esistono licenze [free](#) non [open source](#), e [open source](#) non [free](#)
  - In pratica la stragrande maggioranza dei progetti di [software libero](#) soddisfa i requisiti dell'[open source](#), e viceversa

## Open Source Initiative (OSI)

- 1998: B. Perens e E.S. Raymond fondano la [Open Source Initiative](#)
- Nasce per presentare al mondo industriale e commerciale una visione meno radicale ed idealistica di quella della [Free Software Foundation](#)
- Lo scopo è comunque quello di promuovere software “aperto” che l'utente finale possa studiare ed adattare
- La definizione di [software libero](#) e la definizione di [software open source](#) non coincidono esattamente
  - Esistono licenze [free](#) non [open source](#), e [open source](#) non [free](#)
  - In pratica la stragrande maggioranza dei progetti di [software libero](#) soddisfa i requisiti dell'[open source](#), e viceversa
- L'idea centrale è che la disponibilità del codice sorgente del software
  - aiuta a creare un prodotto migliore grazie allo sviluppo distribuito, ai processi trasparenti ed ai metodi di “peer review”
  - evita alle aziende i meccanismi di “vendor lock-in”
  - può essere commercialmente vantaggioso

## Open Source Initiative (OSI)

- 1998: B. Perens e E.S. Raymond fondano la **Open Source Initiative**
- Nasce per presentare al mondo industriale e commerciale una visione meno radicale ed idealistica di quella della **Free Software Foundation**
- Lo scopo è comunque quello di promuovere software “aperto” che l'utente finale possa studiare ed adattare
- La definizione di **software libero** e la definizione di **software open source** non coincidono esattamente
  - Esistono licenze **free** non **open source**, e **open source** non **free**
  - In pratica la stragrande maggioranza dei progetti di **software libero** soddisfa i requisiti dell'**open source**, e viceversa
- L'idea centrale è che la disponibilità del codice sorgente del software
  - aiuta a creare un prodotto migliore grazie allo sviluppo distribuito, ai processi trasparenti ed ai metodi di “peer review”
  - evita alle aziende i meccanismi di “vendor lock-in”
  - può essere commercialmente vantaggioso

Il “prodotto” di maggior successo: il kernel Linux!

## La nascita del kernel Linux...

- Agosto 1991: appare un nuovo SO per PC Intel 80386

```
From: torvalds@klaava.Helsinki.FI
      (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
```

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 [...]

## La nascita del kernel Linux...

- Agosto 1991: appare un nuovo SO per PC Intel 80386

From: torvalds@klaava.Helsinki.FI  
(**Linus Benedict Torvalds**)  
Newsgroups: comp.os.minix  
Subject: What would you like to see most in minix?  
Summary: small poll for my new operating system  
Date: 25 Aug 91 20:57:08 GMT  
Organization: University of Helsinki

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 [...]

## La nascita del kernel Linux...

- Agosto 1991: appare un nuovo SO per PC Intel 80386

From: torvalds@klaava.Helsinki.FI  
(Linus Benedict Torvalds)  
Newsgroups: comp.os.minix  
Subject: What would you like to see most in minix?  
Summary: small poll for my new operating system  
Date: 25 Aug 91 20:57:08 GMT  
Organization: **University of Helsinki**

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 [...]

## La nascita del kernel Linux...

- Agosto 1991: appare un nuovo SO per PC Intel 80386

From: torvalds@klaava.Helsinki.FI

(Linus Benedict Torvalds)

**Newsgroups: comp.os.minix**

Subject: What would you like to see most in minix?

Summary: small poll for my new operating system

Date: 25 Aug 91 20:57:08 GMT

Organization: University of Helsinki

**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 [...]



## La nascita del kernel Linux...

- Agosto 1991: appare un nuovo SO per PC Intel 80386

```
From: torvalds@klaava.Helsinki.FI
      (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
```

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 [...]

## La nascita del kernel Linux...

- Agosto 1991: appare un nuovo SO per PC Intel 80386

```
From: torvalds@klaava.Helsinki.FI
      (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
```

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 [...]

## La nascita del kernel Linux (2)

- Linus Torvalds era uno studente di informatica, senza (ancora) alcuna seria esperienza di programmazione



## La nascita del kernel Linux (2)

- Linus Torvalds era uno studente di informatica, senza (ancora) alcuna seria esperienza di programmazione
- Inizialmente il progetto di SO era focalizzato su architettura Intel 80386



## La nascita del kernel Linux (2)

- Linus Torvalds era uno studente di informatica, senza (ancora) alcuna seria esperienza di programmazione
- Inizialmente il progetto di SO era focalizzato su architettura Intel 80386
- Interrogato sulla portabilità del progetto, Linus scriveva:



*Simply, I'd say that porting is impossible. It's mostly in C, but most people wouldn't call what I write C. It uses every conceivable feature of the 386 I could find, as it was also a project to teach me about the 386. As already mentioned, it uses a MMU, for both paging (not to disk yet) and segmentation. It's the segmentation that makes it REALLY 386 dependent [Linus Torvalds, comp.os.minix, 25 Aug 1991]*

## La nascita del kernel Linux (2)

- Linus Torvalds era uno studente di informatica, senza (ancora) alcuna seria esperienza di programmazione
- Inizialmente il progetto di SO era focalizzato su architettura Intel 80386
- Interrogato sulla portabilità del progetto, Linus scriveva:



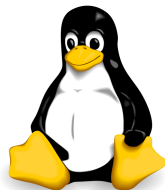
*Simply, I'd say that porting is impossible. It's mostly in C, but most people wouldn't call what I write C. It uses every conceivable feature of the 386 I could find, as it was also a project to teach me about the 386. As already mentioned, it uses a MMU, for both paging (not to disk yet) and segmentation. It's the segmentation that makes it REALLY 386 dependent [Linus Torvalds, comp.os.minix, 25 Aug 1991]*

- All'epoca non esistevano piani per la *conquista del mondo!*



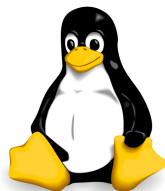
## Il kernel Linux ieri ed oggi

- Nel 1991 (versione 0.01): 76 file, 512 KB su disco, < 8500 linee di codice



## Il kernel Linux ieri ed oggi

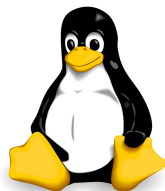
- Nel 1991 (versione 0.01): 76 file, 512 KB su disco, < 8500 linee di codice
- Nel 2013 (versione 3.8-rc6): ~ 42 000 file, 554 MB su disco, 15 310 000 linee di codice





## Il kernel Linux ieri ed oggi

- Nel 1991 (versione 0.01): 76 file, 512 KB su disco, < 8500 linee di codice
- Nel 2013 (versione 3.8-rc6): ~ 42 000 file, 554 MB su disco, 15 310 000 linee di codice

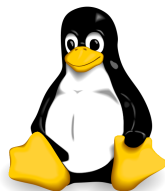


### Le domande cruciali

- Quali sono stati i progressi più importanti nel kernel Linux?

## Il kernel Linux ieri ed oggi

- Nel 1991 (versione 0.01): 76 file, 512 KB su disco, < 8500 linee di codice
- Nel 2013 (versione 3.8-rc6): ~ 42 000 file, 554 MB su disco, 15 310 000 linee di codice

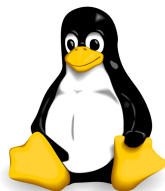


### Le domande cruciali

- Quali sono stati i progressi più importanti nel kernel Linux?
- Chi condivide i meriti della velocissima e continua evoluzione di Linux? È unicamente un successo personale di un programmatore eccezionale come Linus Torvalds?

## Il kernel Linux ieri ed oggi

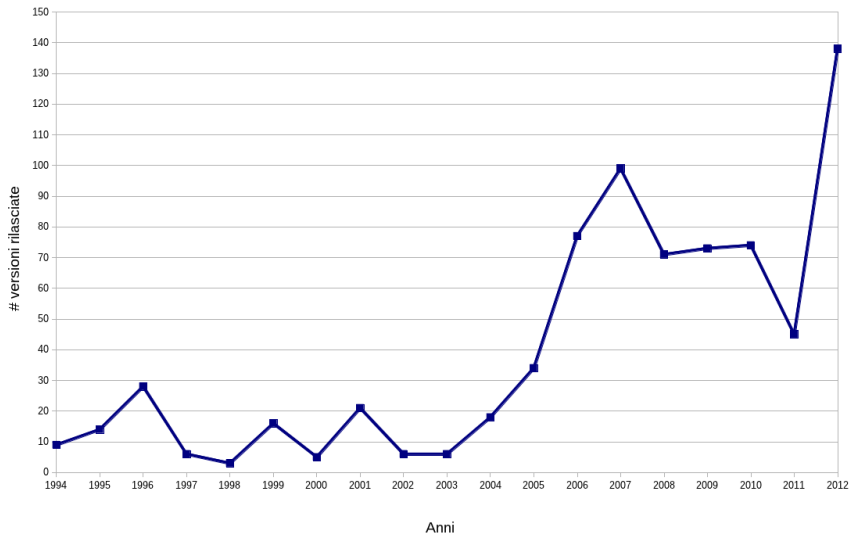
- Nel 1991 (versione 0.01): 76 file, 512 KB su disco, < 8500 linee di codice
- Nel 2013 (versione 3.8-rc6): ~ 42 000 file, 554 MB su disco, 15 310 000 linee di codice



### Le domande cruciali

- Quali sono stati i progressi più importanti nel kernel Linux?
- Chi condivide i meriti della velocissima e continua evoluzione di Linux? È unicamente un successo personale di un programmatore eccezionale come Linus Torvalds?
- Come è accaduto che un SO giocattolo di uno studente di secondo anno di informatica è diventato un SO di successo che compete con quelli delle più importanti aziende di IT?

## Le versioni stabili di Linux



Sono contate anche le versioni minori ("bug fix")

## Dimensione del codice sorgente

- La dimensione del codice sorgente non è di per se un parametro direttamente legato alle qualità e caratteristiche del kernel

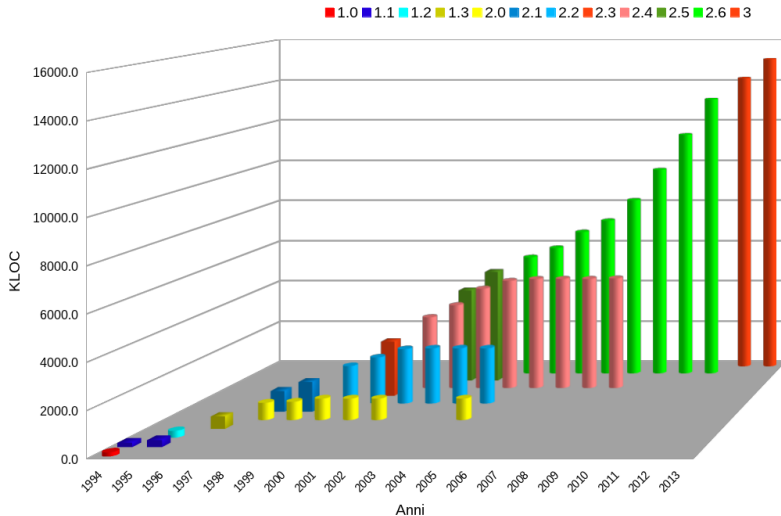
## Dimensione del codice sorgente

- La dimensione del codice sorgente non è di per se un parametro direttamente legato alle qualità e caratteristiche del kernel
- Tuttavia la dimensione del codice sorgente è legata a:
  - la complessità di progetto del kernel
  - la difficoltà di gestione del progetto

## Dimensione del codice sorgente

- La dimensione del codice sorgente non è di per se un parametro direttamente legato alle qualità e caratteristiche del kernel
- Tuttavia la dimensione del codice sorgente è legata a:
  - la complessità di progetto del kernel
  - la difficoltà di gestione del progetto
- Metriche sostanzialmente equivalenti per il kernel Linux:
  - numero di linee di codice
  - numero di funzioni
  - numero di file sorgenti
  - dimensione del codice sorgente su disco

## Dimensione del codice sorgente (2)

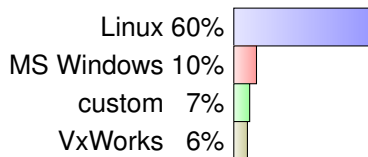




## Diffusione per tipologie di utilizzo

(Stime approssimate ed indicative, da fonti diverse e non omogenee)

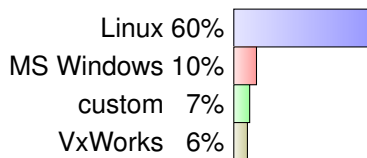
### Sistemi integrati



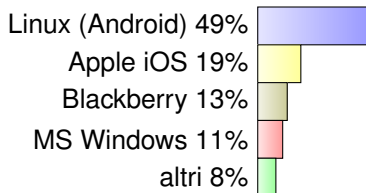
(Stime approssimate ed indicative, da fonti diverse e non omogenee)

## Diffusione per tipologie di utilizzo

### Sistemi integrati



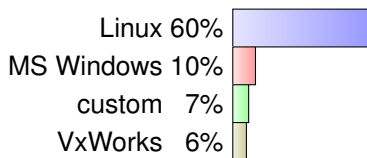
### Mobile (smartphone)



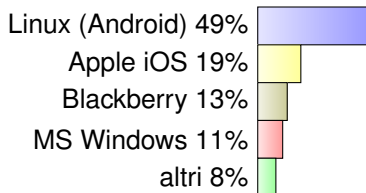
(Stime approssimate ed indicative, da fonti diverse e non omogenee)

## Diffusione per tipologie di utilizzo

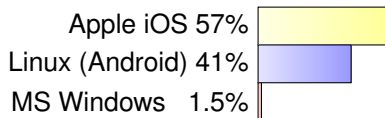
### Sistemi integrati



### Mobile (smartphone)



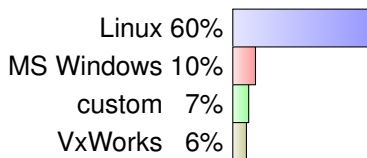
### Tablet



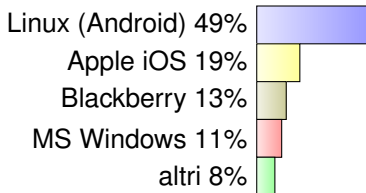
(Stime approssimate ed indicative, da fonti diverse e non omogenee)

## Diffusione per tipologie di utilizzo

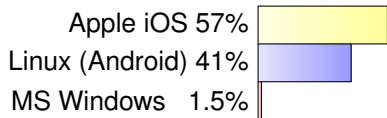
### Sistemi integrati



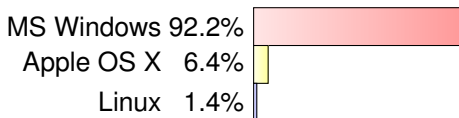
### Mobile (smartphone)



### Tablet



### Laptop & desktop



(Stime approssimate ed indicative, da fonti diverse e non omogenee)

## Diffusione per tipologie di utilizzo (2)

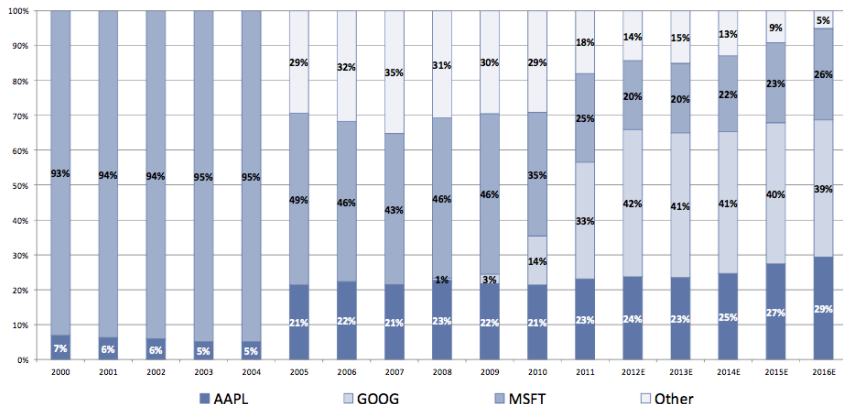
E combinando insieme mobile, tablet, laptop e desktop PC?

## Diffusione per tipologie di utilizzo (2)

E combinando insieme mobile, tablet, laptop e desktop PC?

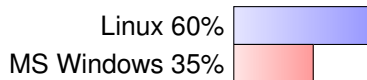
### Exhibit 1: Vendor share of consumer compute, 2000-2016E

Shift from single-vendor dominance (MSFT) to multiple vendors (AAPL, GOOG, MSFT, Other)



Source: IDC, Goldman Sachs Research.

### Server di rete

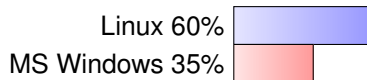


(Stime approssimate ed indicative, da fonti diverse e non omogenee)

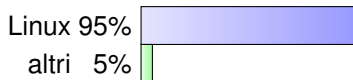


## Diffusione per tipologie di utilizzo (3)

### Server di rete



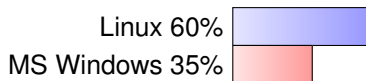
### Mainframe



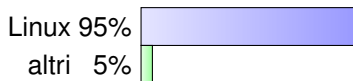
(Stime approssimate ed indicative, da fonti diverse e non omogenee)

## Diffusione per tipologie di utilizzo (3)

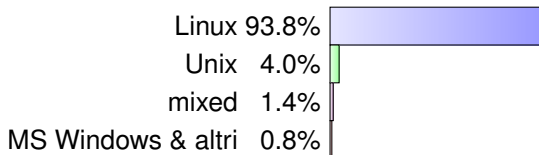
### Server di rete



### Mainframe



### Supercalcolatori (lista TOP500, nov. 2012)



(Stime approssimate ed indicative, da fonti diverse e non omogenee)

# Le chiavi del successo di Linux

## Le domande cruciali

Il successo di Linux sarà duraturo?  
Riuscirà a conquistare il mondo?



# Le chiavi del successo di Linux

## Le domande cruciali

Il successo di Linux sarà duraturo?  
Riuscirà a conquistare il mondo?



## Le ragioni del successo di Linux

- 1 Dammi indietro il tuo codice: la licenza GPL

# Le chiavi del successo di Linux

## Le domande cruciali

Il successo di Linux sarà duraturo?  
Riuscirà a conquistare il mondo?



## Le ragioni del successo di Linux

- 1 Dammi indietro il tuo codice: la licenza GPL
- 2 Il codice cambia rapidamente: flessibilità

# Le chiavi del successo di Linux

## Le domande cruciali

Il successo di Linux sarà duraturo?  
Riuscirà a conquistare il mondo?



## Le ragioni del successo di Linux

- 1 Dammi indietro il tuo codice: la licenza GPL
- 2 Il codice cambia rapidamente: flessibilità
- 3 Responsabilità delegate: gestione intelligente del progetto

# Le chiavi del successo di Linux

## Le domande cruciali

Il successo di Linux sarà duraturo?  
Riuscirà a conquistare il mondo?



## Le ragioni del successo di Linux

- 1 Dammi indietro il tuo codice: la licenza GPL
- 2 Il codice cambia rapidamente: flessibilità
- 3 Responsabilità delegate: gestione intelligente del progetto
- 4 Personalità dei *kernel hacker*: programmatori professionali che amano il loro mestiere

# Le chiavi del successo di Linux

## Le domande cruciali

Il successo di Linux sarà duraturo?  
Riuscirà a conquistare il mondo?



## Le ragioni del successo di Linux

- 1 Dammi indietro il tuo codice: la licenza GPL
- 2 Il codice cambia rapidamente: flessibilità
- 3 Responsabilità delegate: gestione intelligente del progetto
- 4 Personalità dei *kernel hacker*: programmatori professionali che amano il loro mestiere
- 5 Contributi dall'industria: supporto essenziale da tante industrie IT



# Le chiavi del successo di Linux

## Le domande cruciali

Il successo di Linux sarà duraturo?  
Riuscirà a conquistare il mondo?



## Le ragioni del successo di Linux

- 1 Dammi indietro il tuo codice: la licenza GPL
- 2 Il codice cambia rapidamente: flessibilità
- 3 Responsabilità delegate: gestione intelligente del progetto
- 4 Personalità dei *kernel hacker*: programmatori professionali che amano il loro mestiere
- 5 Contributi dall'industria: supporto essenziale da tante industrie IT

Tutte queste ragioni sono fortemente correlate tra loro

## Il progetto GNU e la licenza GPL

- Il successo di Linux è radicato nel progetto GNU di Richard Stallman (*GNU's not UNIX*) per un SO *free* (nel senso della libertà di parola)

## Il progetto GNU e la licenza GPL

- Il successo di Linux è radicato nel progetto GNU di Richard Stallman (*GNU's not UNIX*) per un SO *free* (nel senso della libertà di parola)
- Il progetto GNU ha fornito molti strumenti essenziali per Linux: il compilatore GCC, la libreria standard C, i comandi di sistema, ...

## Il progetto GNU e la licenza GPL

- Il successo di Linux è radicato nel progetto GNU di Richard Stallman (*GNU's not UNIX*) per un SO *free* (nel senso della libertà di parola)
- Il progetto GNU ha fornito molti strumenti essenziali per Linux: il compilatore GCC, la libreria standard C, i comandi di sistema, ...
- La licenza GNU GPL è un ingrediente chiave per il successo di Linux

### La GPL in sintesi

Chiunque può liberamente leggere e modificare il codice sorgente del kernel a condizione che, se il nuovo prodotto è distribuito, il codice sorgente modificato è reso disponibile agli utenti finali ed agli sviluppatori del kernel con la stessa licenza

## Il progetto GNU e la licenza GPL

- Il successo di Linux è radicato nel progetto GNU di Richard Stallman (*GNU's not UNIX*) per un SO *free* (nel senso della libertà di parola)
- Il progetto GNU ha fornito molti strumenti essenziali per Linux: il compilatore GCC, la libreria standard C, i comandi di sistema, ...
- La licenza GNU GPL è un ingrediente chiave per il successo di Linux

### La GPL in sintesi

Chiunque può liberamente leggere e modificare il codice sorgente del kernel a condizione che, se il nuovo prodotto è distribuito, il codice sorgente modificato è reso disponibile agli utenti finali ed agli sviluppatori del kernel con la stessa licenza

- Comunque il successo di Linux è dovuto anche ad altri fattori oltre ai meriti del progetto GNU

## Velocità con cui Linux si evolve

Il codice del kernel viene modificato con un ritmo impressionante:  
una *major release* ogni 2 o 3 mesi

## Velocità con cui Linux si evolve

Il codice del kernel viene modificato con un ritmo impressionante:  
una *major release* ogni 2 o 3 mesi

|                             |  | Versioni | Changeset | Linee di codice<br>aggiunte      modificate |         |
|-----------------------------|--|----------|-----------|---------------------------------------------|---------|
| Statistiche<br>per il 2012: |  | 3.3      | 10 550    | 555 037                                     | 155 356 |
|                             |  | 3.4      | 10 899    | 450 300                                     | 183 339 |
|                             |  | 3.5      | 10 957    | 571 987                                     | 135 848 |
|                             |  | 3.6      | 10 247    | 459 722                                     | 130 435 |
|                             |  | 3.7      | 11 990    | 1 552 034                                   | 147 307 |
|                             |  | Totale   | 54 643    | 3 589 080                                   | 752 285 |

## Velocità con cui Linux si evolve

Il codice del kernel viene modificato con un ritmo impressionante:  
una *major release* ogni 2 o 3 mesi

| Statistiche<br>per il 2012: | Versioni | Changeset | Linee di codice |            |
|-----------------------------|----------|-----------|-----------------|------------|
|                             |          |           | aggiunte        | modificate |
|                             | 3.3      | 10 550    | 555 037         | 155 356    |
|                             | 3.4      | 10 899    | 450 300         | 183 339    |
|                             | 3.5      | 10 957    | 571 987         | 135 848    |
|                             | 3.6      | 10 247    | 459 722         | 130 435    |
|                             | 3.7      | 11 990    | 1 552 034       | 147 307    |
|                             | Totale   | 54 643    | 3 589 080       | 752 285    |

Statisticamente, in ciascuna *ora* dell'anno 2012 gli sviluppatori del kernel hanno aggiunto 410 linee di codice e modificato altre 86 linee!



## Velocità con cui Linux si evolve

Il codice del kernel viene modificato con un ritmo impressionante:  
una *major release* ogni 2 o 3 mesi

| Statistiche<br>per il 2012: | Versioni | Changeset | Linee di codice<br>aggiunte | Linee di codice<br>modificate |
|-----------------------------|----------|-----------|-----------------------------|-------------------------------|
|                             | 3.3      | 10 550    | 555 037                     | 155 356                       |
|                             | 3.4      | 10 899    | 450 300                     | 183 339                       |
|                             | 3.5      | 10 957    | 571 987                     | 135 848                       |
|                             | 3.6      | 10 247    | 459 722                     | 130 435                       |
|                             | 3.7      | 11 990    | 1 552 034                   | 147 307                       |
|                             | Totale   | 54 643    | 3 589 080                   | 752 285                       |

Statisticamente, in ciascuna *ora* dell'anno 2012 gli sviluppatori del kernel hanno aggiunto 410 linee di codice e modificato altre 86 linee!

Ma quanti sono gli sviluppatori di Linux?

## Quanti sono gli sviluppatori?

- Il file CREDITS nel codice sorgente di Linux include circa 500 nomi, ma è largamente incompleto

## Quanti sono gli sviluppatori?

- Il file CREDITS nel codice sorgente di Linux include circa 500 nomi, ma è largamente incompleto
- Si stima che le modifiche del kernel nell'anno 2012 siano state proposte da non meno di 7 000 sviluppatori

## Quanti sono gli sviluppatori?

- Il file CREDITS nel codice sorgente di Linux include circa 500 nomi, ma è largamente incompleto
- Si stima che le modifiche del kernel nell'anno 2012 siano state proposte da non meno di 7 000 sviluppatori
  - in realtà sono di più perché i *team* contano in genere come uno

## Quanti sono gli sviluppatori?

- Il file CREDITS nel codice sorgente di Linux include circa 500 nomi, ma è largamente incompleto
- Si stima che le modifiche del kernel nell'anno 2012 siano state proposte da non meno di 7 000 sviluppatori
  - in realtà sono di più perché i *team* contano in genere come uno
- Si stima che la comunità di sviluppatori di Linux sia oggi composta da circa 8 000 membri

## Quanti sono gli sviluppatori?

- Il file CREDITS nel codice sorgente di Linux include circa 500 nomi, ma è largamente incompleto
- Si stima che le modifiche del kernel nell'anno 2012 siano state proposte da non meno di 7 000 sviluppatori
  - in realtà sono di più perché i *team* contano in genere come uno
- Si stima che la comunità di sviluppatori di Linux sia oggi composta da circa 8 000 membri
- Il numero di sviluppatori è gigantesco:  
come può Linus Torvalds gestire tutti loro?

## Quanti sono gli sviluppatori?

- Il file CREDITS nel codice sorgente di Linux include circa 500 nomi, ma è largamente incompleto
- Si stima che le modifiche del kernel nell'anno 2012 siano state proposte da non meno di 7 000 sviluppatori
  - in realtà sono di più perché i *team* contano in genere come uno
- Si stima che la comunità di sviluppatori di Linux sia oggi composta da circa 8 000 membri
- Il numero di sviluppatori è gigantesco:  
come può Linus Torvalds gestire tutti loro?

“Open-source development violates almost all known management theories”

*[Dr. Marietta Baba, Dean of the Dept. of Social Science, Michigan State Univ.]*



## I responsabili di sottosistema

- Il codice sorgente del kernel è modulare: componenti centrali, driver dei bus, stack di rete, driver di dispositivi sono logicamente separati ed hanno interfacce ben definite



## I responsabili di sottosistema

- Il codice sorgente del kernel è modulare: componenti centrali, driver dei bus, stack di rete, driver di dispositivi sono logicamente separati ed hanno interfacce ben definite
- Linus Torvalds ha nominato diverse centinaia di *responsabili* (*maintainer*): ciascuno è un programmatore con la responsabilità di seguire uno specifico componente o sottosistema del kernel

## I responsabili di sottosistema

- Il codice sorgente del kernel è modulare: componenti centrali, driver dei bus, stack di rete, driver di dispositivi sono logicamente separati ed hanno interfacce ben definite
- Linus Torvalds ha nominato diverse centinaia di *responsabili* (*maintainer*): ciascuno è un programmatore con la responsabilità di seguire uno specifico componente o sottosistema del kernel
- Ogni responsabile coordina un certo numero di programmatori che lavorano sul componente o sottosistema

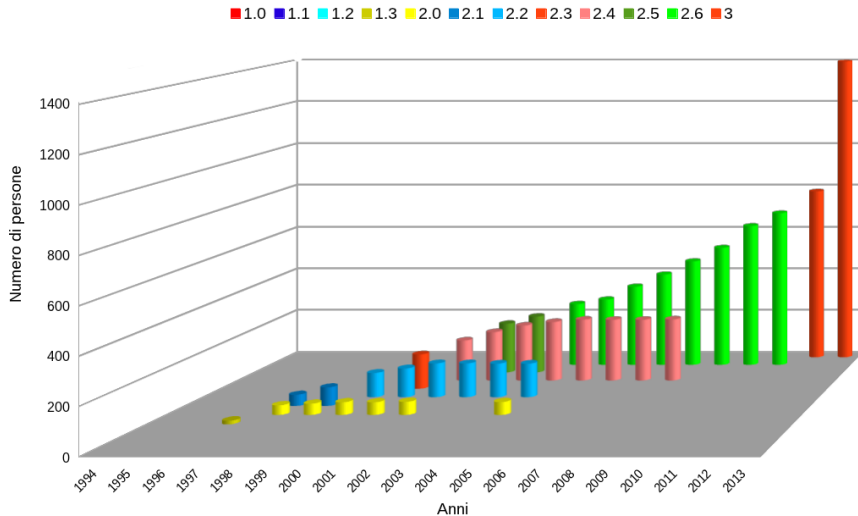
## I responsabili di sottosistema

- Il codice sorgente del kernel è modulare: componenti centrali, driver dei bus, stack di rete, driver di dispositivi sono logicamente separati ed hanno interfacce ben definite
- Linus Torvalds ha nominato diverse centinaia di *responsabili* (*maintainer*): ciascuno è un programmatore con la responsabilità di seguire uno specifico componente o sottosistema del kernel
- Ogni responsabile coordina un certo numero di programmatori che lavorano sul componente o sottosistema
- Come regola generale, le modifiche ad un componente o sottosistema debbono essere proposte a Torvalds dal responsabile interessato

## I responsabili di sottosistema

- Il codice sorgente del kernel è modulare: componenti centrali, driver dei bus, stack di rete, driver di dispositivi sono logicamente separati ed hanno interfacce ben definite
- Linus Torvalds ha nominato diverse centinaia di *responsabili* (*maintainer*): ciascuno è un programmatore con la responsabilità di seguire uno specifico componente o sottosistema del kernel
- Ogni responsabile coordina un certo numero di programmatori che lavorano sul componente o sottosistema
- Come regola generale, le modifiche ad un componente o sottosistema debbono essere proposte a Torvalds dal responsabile interessato
- Alcuni *responsabili generali* hanno compiti particolari:
  - Greg Kroah-Hartman, Chris Wright: fix per le versioni stabili
  - Stephen Rothwell: segue un ramo sperimentale del kernel per il test di nuove funzionalità
  - Andrew Morton: segue un altro ramo sperimentale del kernel dedicato alla gestione della memoria

# Numero di responsabili



## Chi sono gli sviluppatori di Linux

- Linus Torvalds e i responsabili generali non analizzano a fondo ogni singola modifica al kernel, ma solo quelle più critiche (10%–20%)

## Chi sono gli sviluppatori di Linux

- Linus Torvalds e i responsabili generali non analizzano a fondo ogni singola modifica al kernel, ma solo quelle più critiche (10%–20%)
- La robustezza di Linux perciò dipende in larga misura dalla qualità del lavoro di responsabili e programmatori

## Chi sono gli sviluppatori di Linux

- Linus Torvalds e i responsabili generali non analizzano a fondo ogni singola modifica al kernel, ma solo quelle più critiche (10%–20%)
- La robustezza di Linux perciò dipende in larga misura dalla qualità del lavoro di responsabili e programmatori

“[...] it is strange: people who I believe are some of the best programmers in the world choose to work on free software”

*[Andrew Morton, 2005]*





## Chi sono gli sviluppatori di Linux

- Linus Torvalds e i responsabili generali non analizzano a fondo ogni singola modifica al kernel, ma solo quelle più critiche (10%–20%)
- La robustezza di Linux perciò dipende in larga misura dalla qualità del lavoro di responsabili e programmatori

“[...] it is strange: people who I believe are some of the best programmers in the world choose to work on free software”

*[Andrew Morton, 2005]*

“Even though we’re all paid to do kernel work, the culture is largely unchanged from the earliest days of Linux. We work on it because we love our craft and because it allows us to contribute to our society in the way in which we are most able. Corny but true.”

*[Andrew Morton, 2005]*



- In origine Linux era “nelle mani” di una comunità di *computer geeks* che spendevano molto tempo libero scrivendo codice

## Linux e l'industria

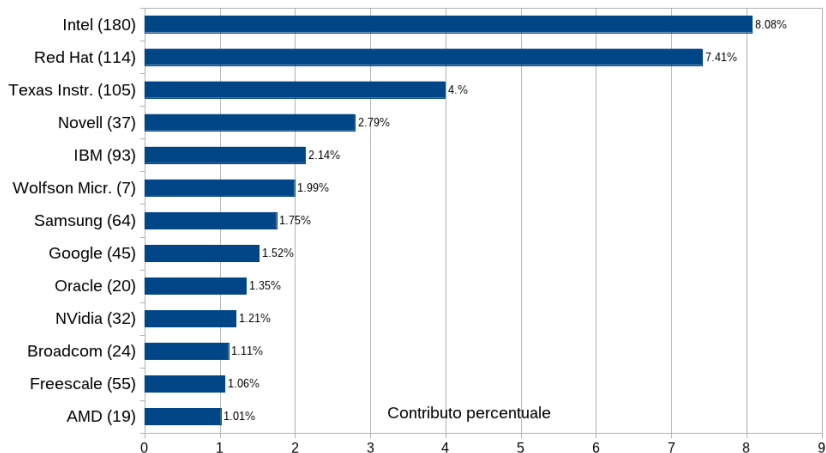
- In origine Linux era “nelle mani” di una comunità di *computer geeks* che spendevano molto tempo libero scrivendo codice
- Oggi questo non è più vero: grandi e piccole aziende di IT supportano Linux assumendo programmatori di prima classe e lasciandoli lavorare a tempo pieno sul kernel

- In origine Linux era “nelle mani” di una comunità di *computer geeks* che spendevano molto tempo libero scrivendo codice
- Oggi questo non è più vero: grandi e piccole aziende di IT supportano Linux assumendo programmatori di prima classe e lasciandoli lavorare a tempo pieno sul kernel
- Ogni anno nel kernel sono integrate le modifiche sviluppate da centinaia di aziende

- In origine Linux era “nelle mani” di una comunità di *computer geeks* che spendevano molto tempo libero scrivendo codice
- Oggi questo non è più vero: grandi e piccole aziende di IT supportano Linux assumendo programmatori di prima classe e lasciandoli lavorare a tempo pieno sul kernel
- Ogni anno nel kernel sono integrate le modifiche sviluppate da centinaia di aziende
- Perché lo fanno? Le ragioni principali:
  - Perché è il loro “*business core*” (es.: Red Hat o Novell)
  - Per garantire che i propri prodotti h/w funzionino a dovere [con Linux]
  - Per adattare e specializzare il kernel Linux in accordo ai propri requisiti

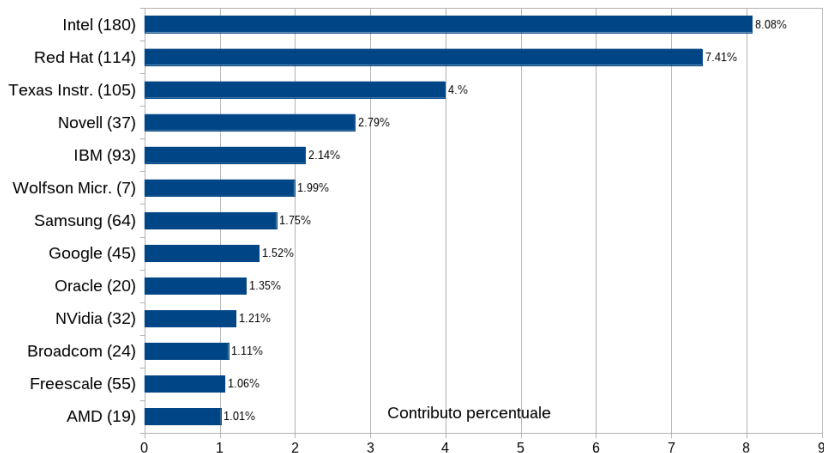
- In origine Linux era “nelle mani” di una comunità di *computer geeks* che spendevano molto tempo libero scrivendo codice
- Oggi questo non è più vero: grandi e piccole aziende di IT supportano Linux assumendo programmatori di prima classe e lasciandoli lavorare a tempo pieno sul kernel
- Ogni anno nel kernel sono integrate le modifiche sviluppate da centinaia di aziende
- Perché lo fanno? Le ragioni principali:
  - Perché è il loro “*business core*” (es.: Red Hat o Novell)
  - Per garantire che i propri prodotti h/w funzionino a dovere [con Linux]
  - Per adattare e specializzare il kernel Linux in accordo ai propri requisiti
- Ciascuna azienda ha un forte interesse nell'ottenere che i propri contributi siano integrati nel codice sorgente ufficiale, così che il proprio codice si evolva di pari passo con il resto del kernel

## Contributi al kernel Linux nel 2012



Tra parentesi il numero di dipendenti dell'azienda che hanno inviato modifiche

## Contributi al kernel Linux nel 2012



Tra parentesi il numero di dipendenti dell'azienda che hanno inviato modifiche

Si stima che circa l'80% dei contributi al kernel provengano dall'industria



## Chi progetta Linux

Poiché le modifiche particolari al kernel Linux sono pilotate dall'industria. . .

Il ruolo principale di Linus Torvalds e degli altri responsabili generali è progettare e pianificare l'evoluzione futura di Linux

## Chi progetta Linux

Poiché le modifiche particolari al kernel Linux sono pilotate dall'industria. . .

~~Il ruolo principale di Linus Torvalds e degli altri responsabili generali è progettare e pianificare l'evoluzione futura di Linux~~

**NO!**

## Chi progetta Linux

Poiché le modifiche particolari al kernel Linux sono pilotate dall'industria. . .

~~Il ruolo principale di Linus Torvalds e degli altri responsabili generali è progettare e pianificare l'evoluzione futura di Linux~~

**NO!**

“Linux is evolution, not intelligent design!” *[Linus Torvalds, 2005]*

## Chi progetta Linux

Poiché le modifiche particolari al kernel Linux sono pilotate dall'industria. . .

~~Il ruolo principale di Linus Torvalds e degli altri responsabili generali è progettare e pianificare l'evoluzione futura di Linux~~

**NO!**

“Linux is evolution, not intelligent design!” *[Linus Torvalds, 2005]*

“People like Linus Torvalds and I don't plan the kernel evolution. We don't sit there and think up the roadmap for the next two years, then assign resources to the various new features. That's because we don't have any resources. The resources are all owned by the various corporations who use and contribute to Linux, as well as by the various independent contributors out there. It's those people who own the resources who decide . . .” *[Andrew Morton, 2005]*

## L'evoluzione di Linux

- La ragione principale del successo di Linux: non è pilotato da qualcuno con un prefissato obiettivo tecnico, commerciale o politico

## L'evoluzione di Linux

- La ragione principale del successo di Linux: non è pilotato da qualcuno con un prefissato obiettivo tecnico, commerciale o politico
- Al contrario, Linux è condizionato dai requisiti delle entità del mondo reale che vogliono adottarlo ed utilizzarlo: aziende di IT e utenti finali

## L'evoluzione di Linux

- La ragione principale del successo di Linux: non è pilotato da qualcuno con un prefissato obiettivo tecnico, commerciale o politico
- Al contrario, Linux è condizionato dai requisiti delle entità del mondo reale che vogliono adottarlo ed utilizzarlo: aziende di IT e utenti finali
- Linus Torvalds è l'“arbitro”: garantisce che ogni modifica al kernel è tecnicamente corretta e, soprattutto, potenzialmente vantaggiosa per l'intera comunità di Linux



# L'evoluzione di Linux

- La ragione principale del successo di Linux: non è pilotato da qualcuno con un prefissato obiettivo tecnico, commerciale o politico
- Al contrario, Linux è condizionato dai requisiti delle entità del mondo reale che vogliono adottarlo ed utilizzarlo: aziende di IT e utenti finali
- Linus Torvalds è l'“arbitro”: garantisce che ogni modifica al kernel è tecnicamente corretta e, soprattutto, potenzialmente vantaggiosa per l'intera comunità di Linux



- Linus Torvalds ha un impiego a tempo pieno nella Linux Foundation: un consorzio “no profit” finanziato da molte grandi aziende di IT e dedicato a promuovere la crescita di Linux

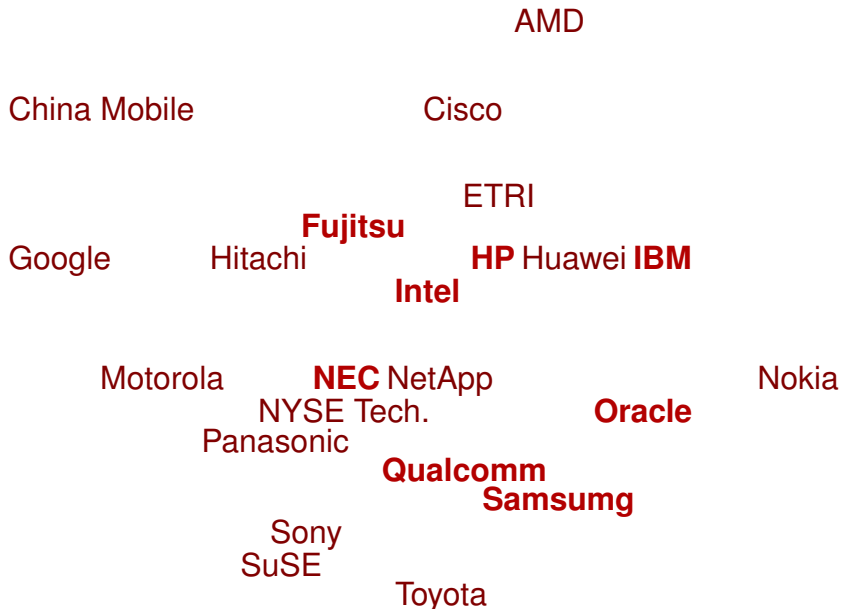


## Chi contribuisce alla Linux Foundation?

A word cloud of company names that contribute to the Linux Foundation. The names are arranged in a roughly circular pattern. The names included are: Fujitsu, HP, IBM, Intel, NEC, Oracle, Qualcomm, and Samsung. The text is in a bold, red, sans-serif font.

Fujitsu  
HP  
IBM  
Intel  
NEC  
Oracle  
Qualcomm  
Samsumg

## Chi contribuisce alla Linux Foundation?



## Chi contribuisce alla Linux Foundation?

4Linux AAVA Mob. Adobe Above Adeneo Emb. **AMD** AISIN AW AllGo Emb. Amarua Sol. Antelink ARM Ashisuto Axis Comm. Barnes & Noble BasysKom Blackduck BORQS Broadcom BT Bull CDAC Calxeda Calix Canonical Cavium Net. **China Mobile** CME Group Cinema **Cisco** Citrix CloudScaling CloudSigma CloudSoft Codero Codethink Collabora Comarch Componentality Credativ CSR Cybercom Group Cypress D-Link DOT Dell Denso Denx DFS DreamHost DreamWorks EMC<sup>2</sup> ENEA EPSON Erlang **ETRI** Eucalyptus Exegy FeuerLabs Fixnetix Fluendo Freescale **Fujitsu** Gazzang Genymobile GoAhead GSmart **Google** Harman **Hitachi** HostConcepts **HP** Huawei **IBM** ICS Igalia Inktank Innominds Interactive Data Intrinsyc **Intel** Ixonos Jaguar Land-Rover J.P.Morgan JVC Kenwood KPT Lanedo LexisNexis Lexmark LG Lineo Linpus LPL Mandriva Marvell Meinberg Mentor Graphics Meyer Sound Micware Mindspeed MIPS Miracle Mocana **Motorola** Nebula **NEC** **NetApp** Nicira Nipa Nissan Nixu **Nokia** NTT NTT Data NVIDIA **NYSE Tech.** OSS Systems **Oracle** OrangeFS OSADL Palamida **Panasonic** Parallels Pelagicore Perforce Pingwinsoft PT ProFUSION Protecode Puzzle ITC **Qualcomm** Reaktor RedHat Renesas Ricoh Rusbitech Savoir-faire Linux Servergy **Samsung** Siemens SIM Tech. Sonatype SprecraCode **Sony** Softlayer Splashtop STEC Superb Internet Symbio ST Symphony Teleca **SuSE** Symbio Synopsys TS-Associates Texas Instr. TMS ThunderSoft Tieto TimeSys Toshiba **Toyota** TurboSystems Tuxera Twitter VirtualBridges VMWare VIA Wind River Yahoo! Zenoss ZTE

## Millennium Technology Prizes

- Dal 2004, il più importante premio per l'innovazione tecnologica

## Millennium Technology Prizes

- Dal 2004, il più importante premio per l'innovazione tecnologica
- Assegnato ogni 2 anni dalla fondazione [Technology Academy Finland](#)

## Millennium Technology Prizes

- Dal 2004, il più importante premio per l'innovazione tecnologica
- Assegnato ogni 2 anni dalla fondazione [Technology Academy Finland](#)
- Primo premio: circa 1 000 000 \$

## Millennium Technology Prizes

- Dal 2004, il più importante premio per l'innovazione tecnologica
- Assegnato ogni 2 anni dalla fondazione [Technology Academy Finland](#)
- Primo premio: circa 1 000 000 \$
- Considerato il “Nobel per la tecnologia”

## Millennium Technology Prizes

- Dal 2004, il più importante premio per l'innovazione tecnologica
- Assegnato ogni 2 anni dalla fondazione [Technology Academy Finland](#)
- Primo premio: circa 1 000 000 \$
- Considerato il “Nobel per la tecnologia”
- Vincitori delle passate edizioni:
  - [2004](#) Tim Berners-Lee (World Wide Web)
  - [2006](#) Shuji Nakamura (LED bianchi e blu)
  - [2008](#) Robert Langer (Biomateriali innovativi)
  - [2010](#) Michael Grätzel (celle solari “dye-sensitized”)



## 2012 Millennium Technology Prize

Vincitori ex-aequo (giugno 2012):

- Linus Torvalds



## 2012 Millennium Technology Prize

Vincitori ex-aequo (giugno 2012):

- Linus Torvalds
- Shinya Yamanaka



## 2012 Millennium Technology Prize

Vincitori ex-aequo (giugno 2012):

- Linus Torvalds
- Shinya Yamanaka



Ottobre 2012: premio Nobel per la medicina assegnato a Sir John B. Gurdon e Shinya Yamanaka per la loro ricerca sulle cellule staminali

“Software is too important in the modern world not to be developed through open source. The real impact of Linux is as a way to allow people and companies to build on top of it to do their own thing. We’re finally getting to the point where “data is just data”, and we don’t have all these insane special communications channels for different forms of data.”



# Grazie!