

Come Linux ha cambiato la mia vita: esperienze di un maturo universitario

Daniel P. Bovet

Università di Roma "Tor Vergata"

bovet@uniroma2.it

Linux Day - Università of Modena - 26 novembre 2005

Preambolo

Preambolo

- Ho vissuto un periodo affascinante dell'informatica che va dall'inizio degli anni '60 ad oggi

Preambolo

- Ho vissuto un periodo affascinante dell'informatica che va dall'inizio degli anni '60 ad oggi
- Vorrei rivivere con voi questo periodo cercando di identificare le principali innovazioni tecnologiche ed i mutamenti sociali che hanno portato al fenomeno Linux

Preambolo

- Ho vissuto un periodo affascinante dell'informatica che va dall'inizio degli anni '60 ad oggi
- Vorrei rivivere con voi questo periodo cercando di identificare le principali innovazioni tecnologiche ed i mutamenti sociali che hanno portato al fenomeno Linux
- Citerò alcuni episodi che mi riguardano ma non li riporterò in questo file: userò appositi capoversi contenenti una dièresi rossa (#) per ricordarmi dove intervenire

Preambolo

- Ho vissuto un periodo affascinante dell'informatica che va dall'inizio degli anni '60 ad oggi
- Vorrei rivivere con voi questo periodo cercando di identificare le principali innovazioni tecnologiche ed i mutamenti sociali che hanno portato al fenomeno Linux
- Citerò alcuni episodi che mi riguardano ma non li riporterò in questo file: userò appositi capoversi contenenti una dieresi rossa (#) per ricordarmi dove intervenire
- #

Di cosa parlerò

Di cosa parlerò

- Di innovazioni hardware e software

Di cosa parlerò

- Di innovazioni hardware e software
- Di come alcuni docenti universitari di sistemi operativi hanno cercato di adattare il loro insegnamento ai tumultuosi cambiamenti degli ultimi decenni

Di cosa parlerò

- Di innovazioni hardware e software
- Di come alcuni docenti universitari di sistemi operativi hanno cercato di adattare il loro insegnamento ai tumultuosi cambiamenti degli ultimi decenni
- Del problema della documentazione del software libero e della documentazione di Linux in particolare

Di cosa parlerò

- Di innovazioni hardware e software
- Di come alcuni docenti universitari di sistemi operativi hanno cercato di adattare il loro insegnamento ai tumultuosi cambiamenti degli ultimi decenni
- Del problema della documentazione del software libero e della documentazione di Linux in particolare
- Di alcune innovazioni, poche a mio parere negative e molte altre positive di Linux 2.6

1959: La nascita dei circuiti integrati

1959: La nascita dei circuiti integrati

- Jack Kilby e Robert Noyce sottomettono due brevetti diversi

1959: La nascita dei circuiti integrati

- Jack Kilby e Robert Noyce sottomettono due brevetti diversi
- Jack Kilby e Texas Instruments ricevono il brevetto U.S. #3,138,743 per circuiti elettronici miniaturizzati

1959: La nascita dei circuiti integrati

- Jack Kilby e Robert Noyce sottomettono due brevetti diversi
- Jack Kilby e Texas Instruments ricevono il brevetto U.S. #3,138,743 per circuiti elettronici miniaturizzati
- Robert Noyce e la Fairchild Semiconductor Corporation ricevono il brevetto U.S. #2,981,877 per un circuito integrato basato sul silicene

1959: La nascita dei circuiti integrati

- Jack Kilby e Robert Noyce sottomettono due brevetti diversi
- Jack Kilby e Texas Instruments ricevono il brevetto U.S. #3,138,743 per circuiti elettronici miniaturizzati
- Robert Noyce e la Fairchild Semiconductor Corporation ricevono il brevetto U.S. #2,981,877 per un circuito integrato basato sul silicene
- Dopo alcuni anni di battaglie legali, le due aziende saggiamente decisero di condividere le loro tecnologie creando così un mercato globale stimato in mille miliardi di US\$ all'anno

1969: La nascita di ARPAnet

1969: La nascita di ARPAnet

- 1963-1965: L. Kleinrock sviluppa un modello probabilistico usato per dimensionare i buffer dei nodi della rete

1969: La nascita di ARPAnet

- 1963-1965: L. Kleinrock sviluppa un modello probabilistico usato per dimensionare i buffer dei nodi della rete
- Nel 1969, ARPAnet collegava i mainframe delle università di Santa Barbara, Utah, UCLA e Stanford

1969: La nascita di ARPAnet

- 1963-1965: L. Kleinrock sviluppa un modello probabilistico usato per dimensionare i buffer dei nodi della rete
- Nel 1969, ARPAnet collegava i mainframe delle università di Santa Barbara, Utah, UCLA e Stanford
- Charley Kline at UCLA sent the first packets on ARPANet as he tried to connect to Stanford Research Institute on Oct 29, 1969. The system crashed as he reached the G in LOGIN!

1969: La nascita di ARPAnet

- 1963-1965: L. Kleinrock sviluppa un modello probabilistico usato per dimensionare i buffer dei nodi della rete
- Nel 1969, ARPAnet collegava i mainframe delle università di Santa Barbara, Utah, UCLA e Stanford
- Charley Kline at UCLA sent the first packets on ARPANet as he tried to connect to Stanford Research Institute on Oct 29, 1969. The system crashed as he reached the G in LOGIN!
- As a graduate student at UCLA, Vint Cerf was involved in the early design of the ARPANET. He was present when the first IMP was delivered to UCLA. He is called the father of the Internet. He earned this nickname as one of the co-authors of TCP/IP

1969: La nascita di ARPAnet

- 1963-1965: L. Kleinrock sviluppa un modello probabilistico usato per dimensionare i buffer dei nodi della rete
- Nel 1969, ARPAnet collegava i mainframe delle università di Santa Barbara, Utah, UCLA e Stanford
- Charley Kline at UCLA sent the first packets on ARPANet as he tried to connect to Stanford Research Institute on Oct 29, 1969. The system crashed as he reached the G in LOGIN!
- As a graduate student at UCLA, Vint Cerf was involved in the early design of the ARPANET. He was present when the first IMP was delivered to UCLA. He is called the father of the Internet. He earned this nickname as one of the co-authors of TCP/IP
- #

1971-1973: La nascita di Unix e di C

1971-1973: La nascita di Unix e di C

- 1971: K. Thompson e D. Ritchie mettono a punto la prima versione di Unix, prima su un PDP 7, poi su un PDP 11/20

1971-1973: La nascita di Unix e di C

- 1971: K. Thompson e D. Ritchie mettono a punto la prima versione di Unix, prima su un PDP 7, poi su un PDP 11/20
- Molte idee del progetto MULTICS dello M.I.T. sono state riprese da Unix

1971-1973: La nascita di Unix e di C

- 1971: K. Thompson e D. Ritchie mettono a punto la prima versione di Unix, prima su un PDP 7, poi su un PDP 11/20
- Molte idee del progetto MULTICS dello M.I.T. sono state riprese da Unix
- 1973: K. Thompson e D. Ritchie riscrivono Unix in C, un linguaggio ad alto livello esplicitamente inventato per scrivere software di base

1974: Il primo simulatore didattico di sistema operativo

1974: Il primo simulatore didattico di sistema operativo

- Il testo “Operating Systems” di S. E. Madnick e J. J. Donovan (McGraw-Hill 1974) riportava in un apposito capitolo il listato assembly di un simulatore di un rudimentale sistema operativo

1974: Il primo simulatore didattico di sistema operativo

- Il testo “Operating Systems” di S. E. Madnick e J. J. Donovan (McGraw-Hill 1974) riportava in un apposito capitolo il listato assembly di un simulatore di un rudimentale sistema operativo
- Gli studenti potevano eseguire prove con tale sistema operativo in modalità batch, ossia sottoponendo pacchi di schede (job) ad un mainframe IBM

1974: Il primo simulatore didattico di sistema operativo

- Il testo “Operating Systems” di S. E. Madnick e J. J. Donovan (McGraw-Hill 1974) riportava in un apposito capitolo il listato assembly di un simulatore di un rudimentale sistema operativo
- Gli studenti potevano eseguire prove con tale sistema operativo in modalità batch, ossia sottoponendo pacchi di schede (job) ad un mainframe IBM
- #

1975: si diffonde il codice sorgente di Unix

1975: si diffonde il codice sorgente di Unix

- AT&T decise ben presto di rilasciare il codice sorgente di Unix ad enti di ricerca ed università

1975: si diffonde il codice sorgente di Unix

- AT&T decise ben presto di rilasciare il codice sorgente di Unix ad enti di ricerca ed università
- Nel 1975 circola clandestinamente un documento fondamentale di John Lions: *Source code and commentary on Unix Level 6*

1975: si diffonde il codice sorgente di Unix

- AT&T decise ben presto di rilasciare il codice sorgente di Unix ad enti di ricerca ed università
- Nel 1975 circola clandestinamente un documento fondamentale di John Lions: *Source code and commentary on Unix Level 6*
- La pubblicazione ufficiale di tale documento viene autorizzata soltanto nel 1996!

1981: Il primo PC IBM

1981: Il primo PC IBM

- Usava un microprocessore Intel 8086 a 16 bit ed utilizzava diversi componenti reperibili sul mercato (architettura non proprietaria)

1981: Il primo PC IBM

- Usava un microprocessore Intel 8086 a 16 bit ed utilizzava diversi componenti reperibili sul mercato (architettura non proprietaria)
- Nel 1982, Compaq introdusse il primo PC “IBM compatibile”

1981: Il primo PC IBM

- Usava un microprocessore Intel 8086 a 16 bit ed utilizzava diversi componenti reperibili sul mercato (architettura non proprietaria)
- Nel 1982, Compaq introdusse il primo PC “IBM compatibile”
- Già a partire dal 1984, numerose aziende producevano PC “IBM compatibile”

1981: Il primo PC IBM

- Usava un microprocessore Intel 8086 a 16 bit ed utilizzava diversi componenti reperibili sul mercato (architettura non proprietaria)
- Nel 1982, Compaq introdusse il primo PC “IBM compatibile”
- Già a partire dal 1984, numerose aziende producevano PC “IBM compatibile”
- Oggi possiamo dare per scontato che tutti i nostri studenti di informatica hanno accesso ad un PC, se non ad un portatile con connessione wireless

1983: nasce il socket programming

1983: nasce il socket programming

- Nel 1980 DARPA assegna a Bolt, Beranek e Newman il compito di implementare lo stack TCP/IP sulla variante BSD di Unix

1983: nasce il socket programming

- Nel 1980 DARPA assegna a Bolt, Beranek e Newman il compito di implementare lo stack TCP/IP sulla variante BSD di Unix
- Nel 1983 viene rilasciato la versione 4.2BSD di Unix che offre l'interfaccia socket

1983: nasce il socket programming

- Nel 1980 DARPA assegna a Bolt, Beranek e Newman il compito di implementare lo stack TCP/IP sulla variante BSD di Unix
- Nel 1983 viene rilasciato la versione 4.2BSD di Unix che offre l'interfaccia socket
- Tale interfaccia, sviluppata inizialmente per il linguaggio C, rappresenta uno standard de facto per il network programming

1983: nasce il socket programming

- Nel 1980 DARPA assegna a Bolt, Beranek e Newman il compito di implementare lo stack TCP/IP sulla variante BSD di Unix
- Nel 1983 viene rilasciato la versione 4.2BSD di Unix che offre l'interfaccia socket
- Tale interfaccia, sviluppata inizialmente per il linguaggio C, rappresenta uno standard de facto per il network programming
- Tutti i moderni sistemi operativi offrono delle API basate sul socket Unix BSD

1984: nasce il progetto GNU

1984: nasce il progetto GNU

- Fin dagli albori dell'informatica scienziati si scambiavano programmi e librerie tramite nastri magnetici o pacchi di schede

1984: nasce il progetto GNU

- Fin dagli albori dell'informatica scienziati si scambiavano programmi e librerie tramite nastri magnetici o pacchi di schede
- Il contributo di Lyons del 1975 fu essenzialmente quello di “trafugare” un sistema operativo creato in ambito industriale verso le università

1984: nasce il progetto GNU

- Fin dagli albori dell'informatica scienziati si scambiavano programmi e librerie tramite nastri magnetici o pacchi di schede
- Il contributo di Lyons del 1975 fu essenzialmente quello di “trafugare” un sistema operativo creato in ambito industriale verso le università
- Lo scopo di GNU ideato da Richard Stallman è invece di realizzare un intero sistema operativo simile a Unix (GNU's Not UNIX) con l'approccio del software libero

1984: nasce il progetto GNU

- Fin dagli albori dell'informatica scienziati si scambiavano programmi e librerie tramite nastri magnetici o pacchi di schede
- Il contributo di Lyons del 1975 fu essenzialmente quello di “trafugare” un sistema operativo creato in ambito industriale verso le università
- Lo scopo di GNU ideato da Richard Stallman è invece di realizzare un intero sistema operativo simile a Unix (GNU's Not UNIX) con l'approccio del software libero
- Il risultato principale fu quello di produrre gcc: un compilatore ottimizzante per diverse piattaforme hardware

1984: nasce il progetto GNU

- Fin dagli albori dell'informatica scienziati si scambiavano programmi e librerie tramite nastri magnetici o pacchi di schede
- Il contributo di Lyons del 1975 fu essenzialmente quello di “trafugare” un sistema operativo creato in ambito industriale verso le università
- Lo scopo di GNU ideato da Richard Stallman è invece di realizzare un intero sistema operativo simile a Unix (GNU's Not UNIX) con l'approccio del software libero
- Il risultato principale fu quello di produrre gcc: un compilatore ottimizzante per diverse piattaforme hardware
- Variants of the GNU operating system, which use the kernel Linux, are now widely used; though these systems are often referred to as Linux, they are more accurately called GNU/Linux system

1986: il primo microprocessore Intel con paginazione

1986: il primo microprocessore Intel con paginazione

- Sistemi operativi multiutente richiedono un circuito di paginazione hardware che consente di proteggere gli spazi degli indirizzi dei vari processi

1986: il primo microprocessore Intel con paginazione

- Sistemi operativi multiutente richiedono un circuito di paginazione hardware che consente di proteggere gli spazi degli indirizzi dei vari processi
- Il microprocessore Intel 80386 consentì di installare su PC a prezzo contenuto sistemi operativi di tipo Unix

1986: il primo microprocessore Intel con paginazione

- Sistemi operativi multiutente richiedono un circuito di paginazione hardware che consente di proteggere gli spazi degli indirizzi dei vari processi
- Il microprocessore Intel 80386 consentì di installare su PC a prezzo contenuto sistemi operativi di tipo Unix
- Prima di tale data, Unix poteva soltanto essere installato su processori dal costo molto più elevato

1986: il primo microprocessore Intel con paginazione

- Sistemi operativi multiutente richiedono un circuito di paginazione hardware che consente di proteggere gli spazi degli indirizzi dei vari processi
- Il microprocessore Intel 80386 consentì di installare su PC a prezzo contenuto sistemi operativi di tipo Unix
- Prima di tale data, Unix poteva soltanto essere installato su processori dal costo molto più elevato
- #

1987: il primo sistema operativo didattico basato su Intel
80386

1987: il primo sistema operativo didattico basato su Intel 80386

- A. S. Tanenbaum realizza Minix, un sistema operativo didattico scritto in C basato su Unix

1987: il primo sistema operativo didattico basato su Intel 80386

- A. S. Tanenbaum realizza Minix, un sistema operativo didattico scritto in C basato su Unix
- Molti docenti di corsi di sistemi operativi hanno adottato Minix come strumento didattico

1987: il primo sistema operativo didattico basato su Intel 80386

- A. S. Tanenbaum realizza Minix, un sistema operativo didattico scritto in C basato su Unix
- Molti docenti di corsi di sistemi operativi hanno adottato Minix come strumento didattico
- Torvalds, il creatore di Linux, è stato uno degli studenti del prof. Tanenbaum

1989: Il primo sistema operativo Unix commerciale basato
su PC

1989: Il primo sistema operativo Unix commerciale basato su PC

- SCO UNIX System V/386

1989: Il primo sistema operativo Unix commerciale basato su PC

- SCO UNIX System V/386
- A differenza di Minix, SCO UNIX System V/386 è un sistema operativo “POSIX-compliant”, ossia un sistema che include tutti i servizi offerti da un moderno Unix System V

1989: Il primo sistema operativo Unix commerciale basato su PC

- SCO UNIX System V/386
- A differenza di Minix, SCO UNIX System V/386 è un sistema operativo “POSIX-compliant”, ossia un sistema che include tutti i servizi offerti da un moderno Unix System V
- La licenza d'uso per tale sistema aveva un costo non indifferente

1991: Partono progetti per realizzare un Unix open source

1991: Partono progetti per realizzare un Unix open source

- freeBSD

1991: Partono progetti per realizzare un Unix open source

- freeBSD
- netBSD

1991: Partono progetti per realizzare un Unix open source

- freeBSD
- netBSD
- openBSD

1991: Partono progetti per realizzare un Unix open source

- freeBSD
- netBSD
- openBSD
- Linux

1994: nasce Linux 1.0.0

1994: nasce Linux 1.0.0

- Linux si impone rispetto ad altri progetti simili per diversi motivi:

1994: nasce Linux 1.0.0

- Linux si impone rispetto ad altri progetti simili per diversi motivi:
 - adotta un modello di sviluppo di tipo “gruppo aperto” anziché “gruppo chiuso”, vedi l’ottimo saggio di E. S. Raymond: “The cathedral and the bazaar”

1994: nasce Linux 1.0.0

- Linux si impone rispetto ad altri progetti simili per diversi motivi:
 - adotta un modello di sviluppo di tipo “gruppo aperto” anziché “gruppo chiuso”, vedi l’ottimo saggio di E. S. Raymond: “The cathedral and the bazaar”
 - fa uso di Internet diventando un progetto che coinvolge programmatori di tutto il mondo

1994: nasce Linux 1.0.0

- Linux si impone rispetto ad altri progetti simili per diversi motivi:
 - adotta un modello di sviluppo di tipo “gruppo aperto” anziché “gruppo chiuso”, vedi l’ottimo saggio di E. S. Raymond: “The cathedral and the bazaar”
 - fa uso di Internet diventando un progetto che coinvolge programmatori di tutto il mondo
 - è splendidamente coordinato dal creatore Linus Torvalds

Anni 90: l'avanzata del software libero

Anni 90: l'avanzata del software libero

- Usi tipici in sistemi Unix proprietari:

Anni 90: l'avanzata del software libero

- Usi tipici in sistemi Unix proprietari:
 - per includere programmi non inclusi nello Unix proprietario: editor particolari, gnuplot, tex, ecc.

Anni 90: l'avanzata del software libero

- Usi tipici in sistemi Unix proprietari:
 - per includere programmi non inclusi nello Unix proprietario: editor particolari, gnuplot, tex, ecc.
 - per sostituire programmi proprietari con altri più efficienti: gcc a posto di cc

Anni 90: l'avanzata del software libero

- Usi tipici in sistemi Unix proprietari:
 - per includere programmi non inclusi nello Unix proprietario: editor particolari, gnuplot, tex, ecc.
 - per sostituire programmi proprietari con altri più efficienti: gcc a posto di cc
 - aggiornamento di pacchetti presenti sul sistema, ma di fatto non proprietari: nuove versioni di sendmail, X11, ecc.

Anni 90: l'avanzata del software libero

- Usi tipici in sistemi Unix proprietari:
 - per includere programmi non inclusi nello Unix proprietario: editor particolari, gnuplot, tex, ecc.
 - per sostituire programmi proprietari con altri più efficienti: gcc a posto di cc
 - aggiornamento di pacchetti presenti sul sistema, ma di fatto non proprietari: nuove versioni di sendmail, X11, ecc.
 - aggiornamento di pacchetti presenti sul sistema, non proprietari, di cui però era a stessa azienda a fornire gli aggiornamenti: nfs

Anni 90: l'avanzata del software libero

- Usi tipici in sistemi Unix proprietari:
 - per includere programmi non inclusi nello Unix proprietario: editor particolari, gnuplot, tex, ecc.
 - per sostituire programmi proprietari con altri più efficienti: gcc a posto di cc
 - aggiornamento di pacchetti presenti sul sistema, ma di fatto non proprietari: nuove versioni di sendmail, X11, ecc.
 - aggiornamento di pacchetti presenti sul sistema, non proprietari, di cui però era a stessa azienda a fornire gli aggiornamenti: nfs
- #

2001-oggi: l'impatto di Linux sulle aziende informatiche

2001-oggi: l'impatto di Linux sulle aziende informatiche

- Diverse aziende hanno smesso di mantenere i loro sistemi Unix proprietari e sono passate a Linux e/o a Windows

2001-oggi: l'impatto di Linux sulle aziende informatiche

- Diverse aziende hanno smesso di mantenere i loro sistemi Unix proprietari e sono passate a Linux e/o a Windows
- La IBM sembra aver scelto di sostituire i suoi sistemi operativi (OS/400, AIX, OS/390) con Linux

2001-oggi: l'impatto di Linux sulle aziende informatiche

- Diverse aziende hanno smesso di mantenere i loro sistemi Unix proprietari e sono passate a Linux e/o a Windows
- La IBM sembra aver scelto di sostituire i suoi sistemi operativi (OS/400, AIX, OS/390) con Linux
- #

2001-oggi: l'impatto di Linux sulle aziende informatiche

- Diverse aziende hanno smesso di mantenere i loro sistemi Unix proprietari e sono passate a Linux e/o a Windows
- La IBM sembra aver scelto di sostituire i suoi sistemi operativi (OS/400, AIX, OS/390) con Linux
- #
- È stata creata nel 2000 l'ente OSDL finanziato da IBM, HP, CA, Intel, e NEC

2001-oggi: l'impatto di Linux sulle aziende informatiche

- Diverse aziende hanno smesso di mantenere i loro sistemi Unix proprietari e sono passate a Linux e/o a Windows
- La IBM sembra aver scelto di sostituire i suoi sistemi operativi (OS/400, AIX, OS/390) con Linux
- #
- È stata creata nel 2000 l'ente OSDL finanziato da IBM, HP, CA, Intel, e NEC
- Il creatore di Linux, Linus Torvalds, lavora attualmente presso OSDL!

Finalità di OSDL

Finalità di OSDL

- To be the recognized center of gravity for Linux; the central body dedicated to accelerating the use of Linux for enterprise computing through:

Finalità di OSDL

- To be the recognized center of gravity for Linux; the central body dedicated to accelerating the use of Linux for enterprise computing through:
 - Enterprise-class testing and other technical support for the Linux development community

Finalità di OSDL

- To be the recognized center of gravity for Linux; the central body dedicated to accelerating the use of Linux for enterprise computing through:
 - Enterprise-class testing and other technical support for the Linux development community
 - Marshalling of Linux-industry resources to focus investment on areas of greatest need thereby eliminating inhibitors to growth

Finalità di OSDL

- To be the recognized center of gravity for Linux; the central body dedicated to accelerating the use of Linux for enterprise computing through:
 - Enterprise-class testing and other technical support for the Linux development community
 - Marshalling of Linux-industry resources to focus investment on areas of greatest need thereby eliminating inhibitors to growth
 - Practical guidance to our members - vendors and end users alike - on working effectively with the Linux development community

Scrivere codice oppure documentarlo?

Scrivere codice oppure documentarlo?

- I programmatori Linux più brillanti ricavano molte più soddisfazioni dal migliorare codice esistente o dall'aggiungere nuove "feature" piuttosto che dal documentare l'architettura software dell'intero progetto

Scrivere codice oppure documentarlo?

- I programmatori Linux più brillanti ricavano molte più soddisfazioni dal migliorare codice esistente o dall'aggiungere nuove "feature" piuttosto che dal documentare l'architettura software dell'intero progetto
- In diversi commenti presenti nel codice si fa spesso uso di un linguaggio per addetti ai lavori, con il chiaro intento di tenere alla larga i neofiti

Scrivere codice oppure documentarlo?

- I programmatori Linux più brillanti ricavano molte più soddisfazioni dal migliorare codice esistente o dall'aggiungere nuove "feature" piuttosto che dal documentare l'architettura software dell'intero progetto
- In diversi commenti presenti nel codice si fa spesso uso di un linguaggio per addetti ai lavori, con il chiaro intento di tenere alla larga i neofiti
- I programmatori Linux più brillanti trovano facilmente lavoro o consulenze presso società che operano nel mondo Linux

Scrivere codice oppure documentarlo?

- I programmatori Linux più brillanti ricavano molte più soddisfazioni dal migliorare codice esistente o dall'aggiungere nuove "feature" piuttosto che dal documentare l'architettura software dell'intero progetto
- In diversi commenti presenti nel codice si fa spesso uso di un linguaggio per addetti ai lavori, con il chiaro intento di tenere alla larga i neofiti
- I programmatori Linux più brillanti trovano facilmente lavoro o consulenze presso società che operano nel mondo Linux
- Non sono a conoscenza di simili gratificazioni per i documentalisti

La documentazione di Linux

La documentazione di Linux

- La documentazione di Linux non è all'altezza del codice

La documentazione di Linux

- La documentazione di Linux non è all'altezza del codice
- La directory `linux/Documentation` acclusa al codice sorgente include due tipi di documentazione con formati diversi che coprono senza pretesa di completezza alcune parti di Linux

La documentazione di Linux

- La documentazione di Linux non è all'altezza del codice
- La directory `linux/Documentation` acclusa al codice sorgente include due tipi di documentazione con formati diversi che coprono senza pretesa di completezza alcune parti di Linux
- Esistono diversi progetti per realizzare una documentazione di Linux software libero di buon livello, primo fra tutti il Linux Documentation project (www.tldp.org), ma nessuno di essi sembra avere centrato l'obiettivo

La documentazione di Linux

- La documentazione di Linux non è all'altezza del codice
- La directory `linux/Documentation` acclusa al codice sorgente include due tipi di documentazione con formati diversi che coprono senza pretesa di completezza alcune parti di Linux
- Esistono diversi progetti per realizzare una documentazione di Linux software libero di buon livello, primo fra tutti il Linux Documentation project (www.tldp.org), ma nessuno di essi sembra avere centrato l'obiettivo
- Una notevole eccezione è il testo online di A. Rubini et al. "Linux Device Drivers" (www.oreilly.com/catalog/linuxdrive3/book/index.csp)

La documentazione di Linux

- La documentazione di Linux non è all'altezza del codice
- La directory `linux/Documentation` acclusa al codice sorgente include due tipi di documentazione con formati diversi che coprono senza pretesa di completezza alcune parti di Linux
- Esistono diversi progetti per realizzare una documentazione di Linux software libero di buon livello, primo fra tutti il Linux Documentation project (www.tldp.org), ma nessuno di essi sembra avere centrato l'obiettivo
- Una notevole eccezione è il testo online di A. Rubini et al. "Linux Device Drivers" (www.oreilly.com/catalog/linuxdrive3/book/index.csp)
- #

L'evoluzione di Linux

L'evoluzione di Linux

- Diventa sempre più mastodontico sia per incorporare nuovi driver, sia per offrire nuovi servizi richiesti dal mondo delle aziende

L'evoluzione di Linux

- Diventa sempre più mastodontico sia per incorporare nuovi driver, sia per offrire nuovi servizi richiesti dal mondo delle aziende
- Linux 2.2.14 (2000) include circa 2 milioni di righe di codice

L'evoluzione di Linux

- Diventa sempre più mastodontico sia per incorporare nuovi driver, sia per offrire nuovi servizi richiesti dal mondo delle aziende
- Linux 2.2.14 (2000) include circa 2 milioni di righe di codice
- Linux 2.4.18 (2002) include circa 4 milioni di righe di codice

L'evoluzione di Linux

- Diventa sempre più mastodontico sia per incorporare nuovi driver, sia per offrire nuovi servizi richiesti dal mondo delle aziende
- Linux 2.2.14 (2000) include circa 2 milioni di righe di codice
- Linux 2.4.18 (2002) include circa 4 milioni di righe di codice
- Linux 2.6.11 (2005) include circa 6 milioni di righe di codice

È possibile insegnare Linux in un corso universitario?

È possibile insegnare Linux in un corso universitario?

- Servono alcuni prerequisiti:

È possibile insegnare Linux in un corso universitario?

- Servono alcuni prerequisiti:
 - un corso di calcolatori con esercizi in assembler

È possibile insegnare Linux in un corso universitario?

- Servono alcuni prerequisiti:
 - un corso di calcolatori con esercizi in assembler
 - un corso di programmazione C

È possibile insegnare Linux in un corso universitario?

- Servono alcuni prerequisiti:
 - un corso di calcolatori con esercizi in assembler
 - un corso di programmazione C
 - un corso introduttivo ai sistemi operativi con particolare riferimento a Unix

È possibile insegnare Linux in un corso universitario?

- Servono alcuni prerequisiti:
 - un corso di calcolatori con esercizi in assembler
 - un corso di programmazione C
 - un corso introduttivo ai sistemi operativi con particolare riferimento a Unix
- Si può impostare un corso, anche trimestrale, in cui discutere il codice di alcuni sottosistemi chiave di Linux quali il VFS, la page cache o lo scheduler

È possibile insegnare Linux in un corso universitario?

- Servono alcuni prerequisiti:
 - un corso di calcolatori con esercizi in assembler
 - un corso di programmazione C
 - un corso introduttivo ai sistemi operativi con particolare riferimento a Unix
- Si può impostare un corso, anche trimestrale, in cui discutere il codice di alcuni sottosistemi chiave di Linux quali il VFS, la page cache o lo scheduler
- #

Decifrare Linux: un nuovo tipo di didattica?

Decifrare Linux: un nuovo tipo di didattica?

- La decifrazione di testi letterari è comunemente praticata nelle facoltà umanistiche

Decifrare Linux: un nuovo tipo di didattica?

- La decifrazione di testi letterari è comunemente praticata nelle facoltà umanistiche
- La decifrazione di programmi scritti da altri è una attività importante, anche se poco praticata

Decifrare Linux: un nuovo tipo di didattica?

- La decifrazione di testi letterari è comunemente praticata nelle facoltà umanistiche
- La decifrazione di programmi scritti da altri è una attività importante, anche se poco praticata
- È complementare alla ingegneria del software

Decifrare Linux: un nuovo tipo di didattica?

- La decifrazione di testi letterari è comunemente praticata nelle facoltà umanistiche
- La decifrazione di programmi scritti da altri è una attività importante, anche se poco praticata
- È complementare alla ingegneria del software
- Nel caso del codice di Linux richiede una ottima conoscenza del C e delle macro del gcc

Decifrare Linux: un nuovo tipo di didattica?

- La decifrazione di testi letterari è comunemente praticata nelle facoltà umanistiche
- La decifrazione di programmi scritti da altri è una attività importante, anche se poco praticata
- È complementare alla ingegneria del software
- Nel caso del codice di Linux richiede una ottima conoscenza del C e delle macro del gcc
- #

Innovazioni negative di Linux 2.6

Innovazioni negative di Linux 2.6

- l'abbandono del concetto di versione stabile

Innovazioni negative di Linux 2.6

- l'abbandono del concetto di versione stabile
- L'introduzione di kobject senza avere prima tentato di classificare in modo organico le varie risorse del computer con conseguenti sovrapposizioni con il proc file system (vedi la infelice strutturazione della directory `/sys`)

Innovazioni positive di Linux 2.6

Innovazioni positive di Linux 2.6

- L'algoritmo di scheduling della CPU che distingue 140 diverse code di priorità e seleziona il "miglior" processo in tempo costante

Innovazioni positive di Linux 2.6

- L'algoritmo di scheduling della CPU che distingue 140 diverse code di priorità e seleziona il "miglior" processo in tempo costante
- La creazione dinamica dei device file in base ai dispositivi registrati riconosciuti (ci avviciniamo alla directory "Risorse del computer" di Windows)

Innovazioni positive di Linux 2.6

- L'algoritmo di scheduling della CPU che distingue 140 diverse code di priorità e seleziona il “miglior” processo in tempo costante
- La creazione dinamica dei device file in base ai dispositivi registrati riconosciuti (ci avviciniamo alla directory “Risorse del computer” di Windows)
- Il kernel preemptive

Innovazioni positive di Linux 2.6

- L'algoritmo di scheduling della CPU che distingue 140 diverse code di priorità e seleziona il "miglior" processo in tempo costante
- La creazione dinamica dei device file in base ai dispositivi registrati riconosciuti (ci avviciniamo alla directory "Risorse del computer" di Windows)
- Il kernel preemptive
- L'algoritmo euristico di read-ahead di file un pò meno empirico del precedente

Innovazioni positive di Linux 2.6

- L'algoritmo di scheduling della CPU che distingue 140 diverse code di priorità e seleziona il "miglior" processo in tempo costante
- La creazione dinamica dei device file in base ai dispositivi registrati riconosciuti (ci avviciniamo alla directory "Risorse del computer" di Windows)
- Il kernel preemptive
- L'algoritmo euristico di read-ahead di file un pò meno empirico del precedente
- #

Innovazioni positive di Linux 2.6

- L'algoritmo di scheduling della CPU che distingue 140 diverse code di priorità e seleziona il “miglior” processo in tempo costante
- La creazione dinamica dei device file in base ai dispositivi registrati riconosciuti (ci avviciniamo alla directory “Risorse del computer” di Windows)
- Il kernel preemptive
- L'algoritmo euristico di read-ahead di file un pò meno empirico del precedente
- #
- La nuova tecnica di sincronizzazione Read-Copy-Update (RCU)