Strumenti per la manipolazione dei pacchetti di rete

GUFIcon 2004

Paolo Pisati

October 5, 2004

- L'integrazione della rete col resto dell'OS

- L'integrazione della rete col resto dell'OS
- Un'idea "strana"...

- L'integrazione della rete col resto dell'OS
- Un'idea "strana"...
- La sua realizzazione

- L'integrazione della rete col resto dell'OS
- Un'idea "strana"...
- La sua realizzazione
- Alcune riflessioni...

socket(AF_INET, SOCK_STREAM, 0)

- Famoso mezzo di IPC

socket(AF_INET, SOCK_STREAM, 0)

- Famoso mezzo di IPC
- Lavorano a livello applicazione

socket(AF_INET, SOCK_STREAM, 0)

- Famoso mezzo di IPC
- Lavorano a livello applicazione
- Solo protocolli supportati

socket(AF_INET, SOCK_RAW, IPPROTO_RAW)

- Medesimo discorso delle socket

socket(AF_INET, SOCK_RAW, IPPROTO_RAW)

- Medesimo discorso delle socket
- Forgiare fino a rete o data link layer

socket(AF_INET, SOCK_RAW, IPPROTO_RAW)

- Medesimo discorso delle socket
- Forgiare fino a rete o data link layer
- Completo controllo (e onere) della comunicazione

socket(AF_INET, SOCK_RAW, IPPROTO_RAW)

- Medesimo discorso delle socket
- Forgiare fino a rete o data link layer
- Completo controllo (e onere) della comunicazione
- Molte volte "pallose" da programmare (libnet ci aiuta)

Berkeley Packet Filter

- Copia pacchetto che matcha il filtro

Berkeley Packet Filter

- Copia pacchetto che matcha il filtro
- Molto diffuso

Berkeley Packet Filter

- Copia pacchetto che matcha il filtro
- Molto diffuso
- Implementazione del protocollo lo deve supportare

Berkeley Packet Filter

- Copia pacchetto che matcha il filtro
- Molto diffuso
- Implementazione del protocollo lo deve supportare
- Non proprio intuitivo...

FreeBSD Netgraph

- Moduli del kernel: estremamente flessibili

FreeBSD Netgraph

- Moduli del kernel: estremamente flessibili
- Estensibili e strutturabili tramite framework (XNF) (chiedete a Satu...)

FreeBSD Netgraph

- Moduli del kernel: estremamente flessibili
- Estensibili e strutturabili tramite framework (XNF) (chiedete a Satu...)
- Difficili e pericolosi da scrivere...

FreeBSD Netgraph

- Moduli del kernel: estremamente flessibili
- Estensibili e strutturabili tramite framework (XNF) (chiedete a Satu...)
- Difficili e pericolosi da scrivere...
- FreeBSD-only

PFIL hook

- Medesimo discorso dei Netgraph

PFIL hook

- Medesimo discorso dei Netgraph
- pfil_add_hook(int (*func)(), void *arg, int flags, struct pfil_head *)

PFIL hook

- Medesimo discorso dei Netgraph
- pfil_add_hook(int (*func)(), void *arg, int flags, struct pfil_head *)
- *BSD li supportano

Feature set sbilanciati...

- Flessibilita'

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"
  - Voglio crearmi delle statistiche particolari

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"
  - Voglio crearmi delle statistiche particolari
  - Voglio fare filtering "nn convenzionale"

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"
  - Voglio crearmi delle statistiche particolari
  - Voglio fare filtering "nn convenzionale"
  - Voglio implementare un mio algoritmo di rete e voglio un ambiente che mi supporti lo sviluppo

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"
  - Voglio crearmi delle statistiche particolari
  - Voglio fare filtering "nn convenzionale"
  - Voglio implementare un mio algoritmo di rete e voglio un ambiente che mi supporti lo sviluppo
- Semplicita'

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"
  - Voglio crearmi delle statistiche particolari
  - Voglio fare filtering "nn convenzionale"
  - Voglio implementare un mio algoritmo di rete e voglio un ambiente che mi supporti lo sviluppo
- Semplicita'
  - Il kernel nn e' proprio semplicissimo...

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"
  - Voglio crearmi delle statistiche particolari
  - Voglio fare filtering "nn convenzionale"
  - Voglio implementare un mio algoritmo di rete e voglio un ambiente che mi supporti lo sviluppo
- Semplicita'
  - Il kernel nn e' proprio semplicissimo...
- Sicurezza

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"
  - Voglio crearmi delle statistiche particolari
  - Voglio fare filtering "nn convenzionale"
  - Voglio implementare un mio algoritmo di rete e voglio un ambiente che mi supporti lo sviluppo
- Semplicita'
  - Il kernel nn e' proprio semplicissimo...
- Sicurezza
  - Un errore nn deve mettere in pericolo la stabilita' della macchina...

Feature set sbilanciati...

- Flessibilita'
  - Che cosa c'e' al byte 12 nel pacchetto o cercare nel payload dei pkt la stringa "freebsd"
  - Voglio crearmi delle statistiche particolari
  - Voglio fare filtering "nn convenzionale"
  - Voglio implementare un mio algoritmo di rete e voglio un ambiente che mi supporti lo sviluppo
- Semplicita'
  - Il kernel nn e' proprio semplicissimo...
- Sicurezza
  - Un errore nn deve mettere in pericolo la stabilita' della macchina...
- Magheggi sempre possibili ma volevo una soluzione definitiva...

- I pacchetti sono solo blocchetti di byte

- I pacchetti sono solo blocchetti di byte
- Etichette per metadati e poi dati (testo dei proto o dati)

- I pacchetti sono solo blocchetti di byte
- Etichette per metadati e poi dati (testo dei proto o dati)
- uint8_t data[len] (array)

- I pacchetti sono solo blocchetti di byte
- Etichette per metadati e poi dati (testo dei proto o dati)
- uint8_t data[len] (array)
- Quale'e' lo strumento che ci permette di manipolare in qualsiasi modo un array?

- I pacchetti sono solo blocchetti di byte
- Etichette per metadati e poi dati (testo dei proto o dati)
- uint8_t data[len] (array)
- Quale'e' lo strumento che ci permette di manipolare in qualsiasi modo un array?
- Una cpu!

- I pacchetti sono solo blocchetti di byte
- Etichette per metadati e poi dati (testo dei proto o dati)
- uint8_t data[len] (array)
- Quale'e' lo strumento che ci permette di manipolare in qualsiasi modo un array?
- Una cpu!
- Scrivere una macchina virtuale che giri all'interno dello stack di rete e che esegua programmi che impostiano noi dall'esterno...

- Completo controllo su tutto il processo

- Completo controllo su tutto il processo
- Sicurezza di un ambiente schermato (sandbox)

- Completo controllo su tutto il processo
- Sicurezza di un ambiente schermato (sandbox)
- Macro istruzioni (create_ip_pkt() e checksum_ip())

- Completo controllo su tutto il processo
- Sicurezza di un ambiente schermato (sandbox)
- Macro istruzioni (create_ip_pkt() e checksum_ip())
- Strutture di appoggio (maschere, timer&c, e strutture dati)

- Completo controllo su tutto il processo
- Sicurezza di un ambiente schermato (sandbox)
- Macro istruzioni (create_ip_pkt() e checksum_ip())
- Strutture di appoggio (maschere, timer&c, e strutture dati)
- Mille altri aspetti...

- Completo controllo su tutto il processo
- Sicurezza di un ambiente schermato (sandbox)
- Macro istruzioni (create_ip_pkt() e checksum_ip())
- Strutture di appoggio (maschere, timer&c, e strutture dati)
- Mille altri aspetti...
- Say hello to NetCPU! :)

- Micro-istruzioni nei fw...

- Micro-istruzioni nei fw...
- Una doccia fredda: BPF

- Micro-istruzioni nei fw...
- Una doccia fredda: BPF
- Sviluppo?

- Micro-istruzioni nei fw...
- Una doccia fredda: BPF
- Sviluppo?
- Quasi inusabile dal punto di vista della cpu...

Accetta IP solo da 128.3.112.15 a 128.3.112.35

```
struct bpf_insn insns[] = {
BPF_STMT(BPF_LD+BPF_H+BPF_ABS, 12),
BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, ETHERTYPE_IP, 0,
8), BPF_STMT(BPF_LD+BPF_W+BPF_ABS, 26),
BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, 0x8003700f, 0, 2),
BPF_STMT(BPF_LD+BPF_W+BPF_ABS, 30),
BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, 0x80037023, 3, 4),
BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, 0x80037023, 0, 3),
BPF_STMT(BPF_LD+BPF_W+BPF_ABS, 30),
BPF_JUMP(BPF_JMP+BPF_JEQ+BPF_K, 0x8003700f, 0, 1),
BPF_STMT(BPF_RET+BPF_K, (u_int)-1),
BPF_STMT(BPF_RET+BPF_K, 0), };
```

- BPF applica il concetto di CPU alla manipolazione dei pkt

- BPF applica il concetto di CPU alla manipolazione dei pkt
- NetCPU utilizza la cpu virtuale come collante per il resto dell'ambiente...

- BPF applica il concetto di CPU alla manipolazione dei pkt
- NetCPU utilizza la cpu virtuale come collante per il resto dell'ambiente...
- Obbiettivi diversi: cpu vs ambiente

- BPF applica il concetto di CPU alla manipolazione dei pkt
- NetCPU utilizza la cpu virtuale come collante per il resto dell'ambiente...
- Obbiettivi diversi: cpu vs ambiente
- Ampi margini di miglioramento nel progetto...

- BPF applica il concetto di CPU alla manipolazione dei pkt
- NetCPU utilizza la cpu virtuale come collante per il resto dell'ambiente...
- Obbiettivi diversi: cpu vs ambiente
- Ampi margini di miglioramento nel progetto...
- "Interprete testo" poco performante: bytecode

- BPF applica il concetto di CPU alla manipolazione dei pkt
- NetCPU utilizza la cpu virtuale come collante per il resto dell'ambiente...
- Obbiettivi diversi: cpu vs ambiente
- Ampi margini di miglioramento nel progetto...
- "Interprete testo" poco performante: bytecode
- Assemblatore e cpu nel kernel

- Nasce NetCPU (Gennaio 2004)

- Nasce NetCPU (Gennaio 2004)
- GufiCON call for papers: inizio dei lavori (Agosto 2004)

- Nasce NetCPU (Gennaio 2004)
- GufiCON call for papers: inizio dei lavori (Agosto 2004)
- MILESTONE: funziona! (19 Settembre 2004)

- Nasce NetCPU (Gennaio 2004)
- GufiCON call for papers: inizio dei lavori (Agosto 2004)
- MILESTONE: funziona! (19 Settembre 2004)
- 2/3 Ottobre GUFICon: :)

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param
- 3 modi di indirizzamento: immediato, relativo e indiretto

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param
- 3 modi di indirizzamento: immediato, relativo e indiretto
- Accesso in lettura/scrittura pacchetto (solo ro interessante per ora)

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param
- 3 modi di indirizzamento: immediato, relativo e indiretto
- Accesso in lettura/scrittura pacchetto (solo ro interessante per ora)
- Pipeline a 3 stadi (fetch, decode e execute)

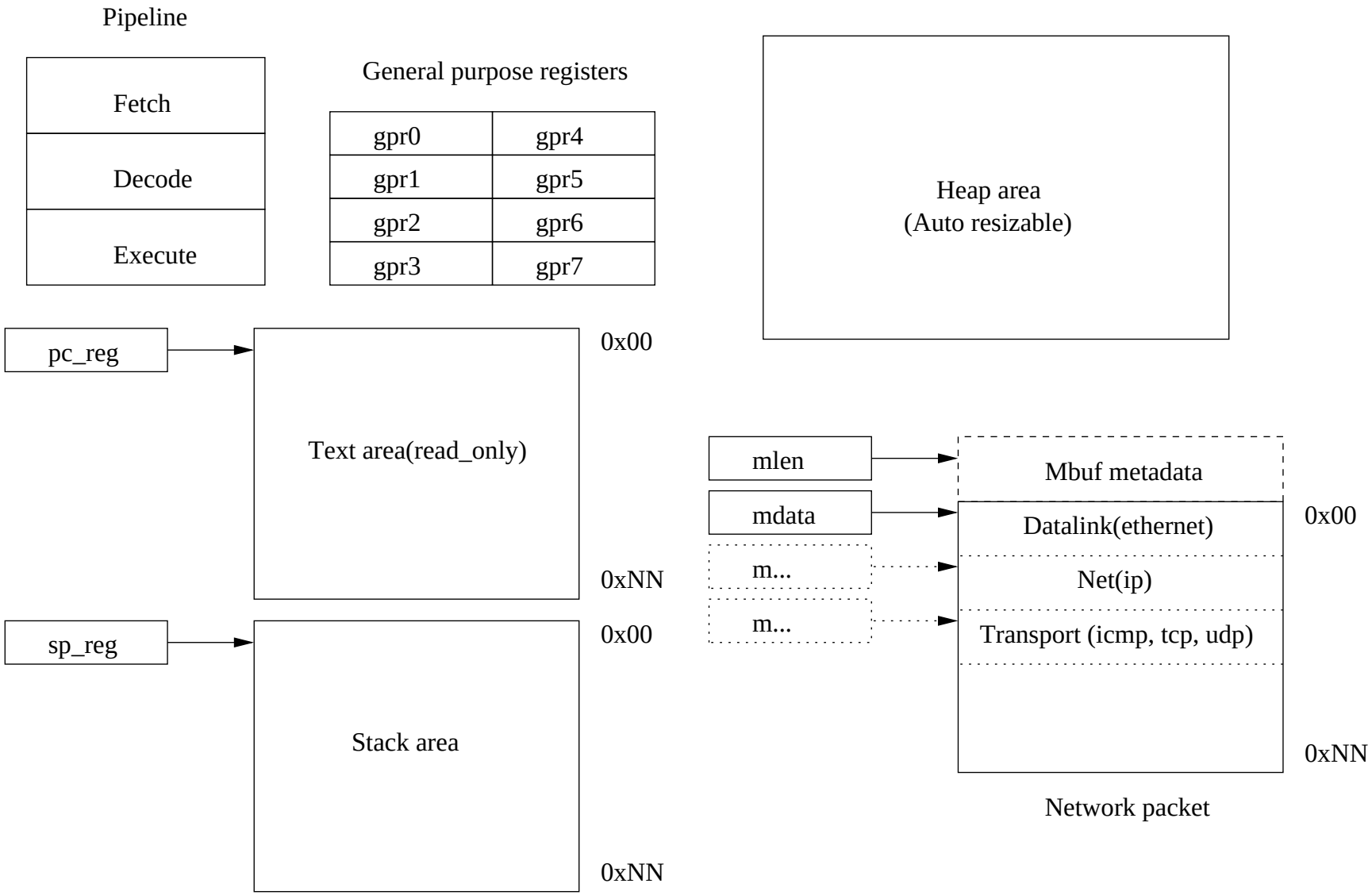
- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param
- 3 modi di indirizzamento: immediato, relativo e indiretto
- Accesso in lettura/scrittura pacchetto (solo ro interessante per ora)
- Pipeline a 3 stadi (fetch, decode e execute)
- Supporto heap ridimensionabile e stack

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param
- 3 modi di indirizzamento: immediato, relativo e indiretto
- Accesso in lettura/scrittura pacchetto (solo ro interessante per ora)
- Pipeline a 3 stadi (fetch, decode e execute)
- Supporto heap ridimensionabile e stack
- Funzioni

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param
- 3 modi di indirizzamento: immediato, relativo e indiretto
- Accesso in lettura/scrittura pacchetto (solo ro interessante per ora)
- Pipeline a 3 stadi (fetch, decode e execute)
- Supporto heap ridimensionabile e stack
- Funzioni
- Variabili nello heap e comportamento statico

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param
- 3 modi di indirizzamento: immediato, relativo e indiretto
- Accesso in lettura/scrittura pacchetto (solo ro interessante per ora)
- Pipeline a 3 stadi (fetch, decode e execute)
- Supporto heap ridimensionabile e stack
- Funzioni
- Variabili nello heap e comportamento statico
- Registri che puntano a parti del pkt (mappagio blind)

- Supporto int32_t e stringhe (nn ancora implementata l'ultima...)
- 8 registri general purpose da 32 bit (prima erano 32...)
- Istruzioni lunghezza fissa 32 bit, 8 max num param
- 3 modi di indirizzamento: immediato, relativo e indiretto
- Accesso in lettura/scrittura pacchetto (solo ro interessante per ora)
- Pipeline a 3 stadi (fetch, decode e execute)
- Supporto heap ridimensionabile e stack
- Funzioni
- Variabili nello heap e comportamento statico
- Registri che puntano a parti del pkt (mappagio blind)
- Supporto costrutti alto livello (come switch)


Pipeline
Fetch
Decode
Execute
General purpose registers
gpr0
gpr4
gpr1
gpr5
gpr2
gpr6
gpr3
gpr7
Heap area
(Auto resizable)
pc_reg
Text area(read_only)
0x00
0xNN
sp_reg
Stack area
0x00
0xNN
mlen
mdata
m...
m...
Mbuf metadata
Datalink(ethernet)
Net(ip)
Transport (icmp, tcp, udp)
0x00
0xNN
Network packet

```
# zona dati nello heap
.data:
num cnt 0 # contatore
#codice parte qui
.text:
mov $gpr0, $mlen # sposto lunghezza pkt in reg
print $gpr0 # stampo lunghezza mbuf
call $mycnt # chiamata a funzione
print 666
end
# la prox e' una funzione
func $mycnt:
add $cnt, 1
ret
```

Alcune note sulla sintassi

- Sezione .init facoltativa

Alcune note sulla sintassi

- Sezione .init facoltativa
- Eliminazione della wildcard $ (registri mappati)

Alcune note sulla sintassi

- Sezione .init facoltativa
- Eliminazione della wildcard $ (registri mappati)
- Supporto rappresentazione hex e binaria

Set di istruzioni

- MOV
- PRINT
- END
- ADD
- SUB
- MUL
- DIV
- CMP
- JE
- JNE
- JMP
- PUSH
- POP
- NOP

CALL
RET
SWITCH
ENDSWITCH
CASE
BREAK
DEFAULT
DROP
AND
OR
XOR
NOT
SHL
SHR
DUMP
LOG

Ambiente

- 2 binari: un programma in userland ed il modulo kernel

Ambiente

- 2 binari: un programma in userland ed il modulo kernel
- Piccola patch per il comando sysctl (aggiunto flag -f)

Ambiente

- 2 binari: un programma in userland ed il modulo kernel
- Piccola patch per il comando sysctl (aggiunto flag -f)
- Comunicazione avviene tramite 3 sysctl

Ambiente

- 2 binari: un programma in userland ed il modulo kernel
- Piccola patch per il comando sysctl (aggiunto flag -f)
- Comunicazione avviene tramite 3 sysctl
- Utilizzo di minibsd e vmware per i test

Ambiente

- 2 binari: un programma in userland ed il modulo kernel
- Piccola patch per il comando sysctl (aggiunto flag -f)
- Comunicazione avviene tramite 3 sysctl
- Utilizzo di minibsd e vmware per i test
- Piccola demo...

- Ripulitura della sintassi dell'asm e del core

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?
  - Nuovo modo di indirizzamento?

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?
  - Nuovo modo di indirizzamento?
- Supporto nativo stringhe

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?
  - Nuovo modo di indirizzamento?
- Supporto nativo stringhe
  - Supporto nativo? Registri stringhe? (Parrot)

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?
  - Nuovo modo di indirizzamento?
- Supporto nativo stringhe
  - Supporto nativo? Registri stringhe? (Parrot)
  - Istruzioni particolari?

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?
  - Nuovo modo di indirizzamento?
- Supporto nativo stringhe
  - Supporto nativo? Registri stringhe? (Parrot)
  - Istruzioni particolari?
- Implementazione dei timer e watchdog

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?
  - Nuovo modo di indirizzamento?
- Supporto nativo stringhe
  - Supporto nativo? Registri stringhe? (Parrot)
  - Istruzioni particolari?
- Implementazione dei timer e watchdog
- Varie&eventuali...

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?
  - Nuovo modo di indirizzamento?
- Supporto nativo stringhe
  - Supporto nativo? Registri stringhe? (Parrot)
  - Istruzioni particolari?
- Implementazione dei timer e watchdog
- Varie&eventuali...
- Test

- Ripulitura della sintassi dell'asm e del core
- Miglior supporto per l'accesso al pacchetto di rete (sicurezza e byte order)
  - Istruzioni particolari (NMOV)?
  - Nuovo modo di indirizzamento?
- Supporto nativo stringhe
  - Supporto nativo? Registri stringhe? (Parrot)
  - Istruzioni particolari?
- Implementazione dei timer e watchdog
- Varie&eventuali...
- Test
- Read-only completo

- Miglior supporto per la scrittura (create_ip_pkt()/checksum())

- Miglior supporto per la scrittura (create_ip_pkt()/checksum())
- PFIL

- Miglior supporto per la scrittura (create_ip_pkt()/checksum())
- PFIL
- Big endian & 64bit & SMP(?!?!)

- Miglior supporto per la scrittura (create_ip_pkt()/checksum())
- PFIL
- Big endian & 64bit & SMP(?!?!)
- NetBSD, OpenBSD e Darwin(PFIL?!?!?)

- Protocollo NetCPU (over IP magari)

- Protocollo NetCPU (over IP magari)
  - aggiornamento parti della cpu...

- Protocollo NetCPU (over IP magari)
  - aggiornamento parti della cpu...
  - od un programma nuovo completo...

- Protocollo NetCPU (over IP magari)
  - aggiornamento parti della cpu...
  - od un programma nuovo completo...
- Struttura dati embedded per la flow classification

- Protocollo NetCPU (over IP magari)
  - aggiornamento parti della cpu...
  - od un programma nuovo completo...
- Struttura dati embedded per la flow classification
- Macroistruzioni (create_pkt()&C)

- Protocollo NetCPU (over IP magari)
  - aggiornamento parti della cpu...
  - od un programma nuovo completo...
- Struttura dati embedded per la flow classification
- Macroistruzioni (create_pkt()&C)
- Ripensare al linguaggio...

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?
  - Dettaglio implementativo?

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?
  - Dettaglio implementativo?
- Performance?

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?
  - Dettaglio implementativo?
- Performance?
  - Effetto cache su core cpu

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?
  - Dettaglio implementativo?
- Performance?
  - Effetto cache su core cpu
  - No copie e context switch per manipolazione dati

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?
  - Dettaglio implementativo?
- Performance?
  - Effetto cache su core cpu
  - No copie e context switch per manipolazione dati
  - Compressione/selezione dei dati da spostare

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?
  - Dettaglio implementativo?
- Performance?
  - Effetto cache su core cpu
  - No copie e context switch per manipolazione dati
  - Compressione/selezione dei dati da spostare
- Pensierino di chiusura: macchine virtuali nel kernel...

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?
  - Dettaglio implementativo?
- Performance?
  - Effetto cache su core cpu
  - No copie e context switch per manipolazione dati
  - Compressione/selezione dei dati da spostare
- Pensierino di chiusura: macchine virtuali nel kernel...
- Profiler di sistema...

- Non ho trovato nulla in letteratura che parlasse di macchine virtuali embedded nello stack di rete...
  - Filone chiuso?
  - Dettaglio implementativo?
- Performance?
  - Effetto cache su core cpu
  - No copie e context switch per manipolazione dati
  - Compressione/selezione dei dati da spostare
- Pensierino di chiusura: macchine virtuali nel kernel...
- Profiler di sistema...
- ...Sun DTRACE!

- Dimostrazione di fattibilita'...

- Dimostrazione di fattibilita'...
- Mi piacerebbe avere del feedback: flag@gufi.org

- Dimostrazione di fattibilita'...
- Mi piacerebbe avere del feedback: flag@gufi.org
- Domande? :)

- Nextie per l'aiuto con VMWARE
- Miz per l'idea del protoccolo NetCPU
- Gmarco per il cavo cross
- Gigi Sullivan per la consulenza sulle socket raw...
- Il "ganassa" perche' e' il "ganassa"! :)
- Il mio coniglio Gauss per avermi fatto compagnia durante le lunghe notti insonni... :)