Formulario esercizio 1 – Dischi e I-Node

Calcolo Blocchi

blocchiPerSuperficie = cilindriPerDIsco * settoriPerDisco

blocchiPerDisco = blocchiPerSuperficie * TestinePerDisco

blocchiPerDisco = cilindriPerDIsco * settoriPerDisco * TestinePerDisco

blocchi = cilindri * testine * settori

Tempo di lettura

TempoDiLettura = TempoDIRotazione + TempoDiRicerca + TempoDiAccesso

RateDiLettura = DimensioneTrasferimento / TempoDiLettura

Capacità Disco

CapacitàTraccia = SettoriPerTraccia * BytePerSettore

CapacitàCilindro = CapacitàTraccia * TraccePerCilindro

TraccePerCilindro = TraccePerSuperficie * NumeroDiTestine

CapacitàDisco = CapacitàCilindro * CilindriPerDisco

CilindriPerDisco = TracciePerSuperficie

Tempo di ricerca - SeekTime

Movimento della testina sul cilindro

- 5ms track-to-track
- 50ms max seek time
- 30ms avg seek time

Hs = tempo richiesto perché la testina inizi a muoversi
Ht = track-to-track

Seek(n) = Hs + n * Ht

Latenza Rotazionale

Circa 5000 – 7000 RPM, 12/8 ms per rivoluzione RPM/60 = RPS
Latenza Massima = 1/RPS = sec/rotazione

Latenza Media = Latenza Massima / 2

Tempo di trasferimento

$$Tempo\ di\ trasferimento\ = \frac{settori\ da\ trasferire}{settori\ in\ una\ traccia} * \ tempo\ di\ rotazione$$

i-node (ext2)

12 puntatori a 12 nodi fisici

1 puntatore a un Blocco Indiretto di Primo livello

1 puntatore a un Blocco Indiretto di Secondo livello

1 puntatore ad un Blocco Indiretto di Terzo livello

Blocco indiretto di primo livello

Dimensione standard 1KB (1024B), indirizzi di 4B

Indirizza 1024/4 = 256 blocchi fisici

Blocco indiretto di secondo livello

Dimensione standard 1KB (1024B), indirizzi di 4B

Indirizza 1024/4 = 256 blocchi indiretti di primo livello

256*256 blocchi fisici

Blocco indiretto di Terzo livello

Dimensione standard 1KB (1024B), indirizzi di 4B

Indirizza 1024/4 = 256 blocchi indiretti di secondo livello

256*256 blocchi fisici

Tutti i blocchi che non sono blocchi fisici sono blocchi di overhead

Shell

[-f nomefile] verifica esistenza di un file, 0 true, 1 false

var1=”test” crea variabile

unset var1 distrugge variabile

export var1 rende variabile globale

$ (sostituzione di variabile) $var1 accede al contenuto di var1

{} espansione multipla

cp nomefile.{txt,bkp} equivale a cp nomefile.txt nomefile.bkp

ls (list files)

```
-A      do not list implied . and ..
-R      recursive
-h      human readable
-l      listing
-S      sort by filesize
```

rm (remove files)

```
-d    remove empty dir
-r    recursive
```

cp (copy files)

```
-r    recursive
--parents   use full source file name under DIRECTORY
```

sort (sort lines of text files)

```
-f    case insensitive
-g    general numeric sort
-h    human readable numbers
-r    reverse order
-t    field separator
-k    field sorting
```

ps (process info)

-e	Select all processes. Identical to **-A**.
-l	Long format.
-F	Extra full format

grep (print lines that match patterns)

-i ignore case

-e ‘’ match regex

-v invert match

-c only print count of match

-a Process a binary file as if it were text

-r recursive

Repetition Operator

? The preceding item is optional and matched at most once.

* The preceding item will be matched zero or more times.

+ The preceding item will be matched one or more times.

{n} The preceding item is matched exactly n times.

{n,} The preceding item is matched n or more times.

{,m} The preceding item is matched at most m times.

{n,m} The preceding item is matched at least n times, but not more than m times.

head (Print the first 10 lines)

-n NUM display first NUM row

-n **-NUM** print all but the last NUM lines of each file

tail (Print the last 10 lines)

-n NUM display last NUM row

-n **+NUM** print all from the NUM° line

cut (remove sections from each line of files)

```
-d    specify field delimiter
--complement      revert match
-f    specify field
      N     select N field
      N-    from field N included to the end
      N-M   from field N to field M both included
      -N    from start to field N included
```

wc (Print newline, word, and byte counts for each FILE)

```
-l    print the newline counts
```

du (Summarize disk usage of the set of FILEs, recursively for directories.)

```
-0    end each output line with NUL
-a    write counts for all files, not just directories
-b    size in Byte
-h    human readable
-s    summarize, print only total
--exclude=PATTERN      exclude files that match PATTERN
```

find (search for files in a directory hierarchy)

find [start dir] [expression]

```
-type  match type [ b,c,d,l,p,f,s ]
-printf “format string”
      %p     File's name.
      %s     File's size in bytes.
      %k     The amount of disk space used for this file in 1 KB blocks
      %i     File's inode number (in decimal).
      %u     File's user name
      %c     File's last status change
      %Ck    File's last status change time in the format specified by k
      %Ak    File's last access time in the format specified by k
-print0     print NULL at the end of each line
-exec command {} \;
```

date (print or set the system date and time)

-d=’@123456’ set date

Date ‘+%flag’ print formatted date

Flags (more on man date)

%s epoch in seconds

%c locale's date and time (e.g., Thu Mar 3 23:05:25 2005)

%D date; same as %m/%d/%y

%T time; same as %H:%M:%S

%d day of month (e.g., 01)

%m month (01..12)

%Y year

%H hour (00..23)

%M minute (00..59)

%S second (00..60)

xargs

-0 Input items are terminated by a null character

-d delim Input items are terminated by the specified character.

-I {} Replace occurrences of {} with names read from standard input.

tar – archiviazione

tar -rvf <archive_name>.tar <files>

awk ‘{ commands } END { post processing }’

sed

find . -type f | grep "02x" | xargs -I {} echo 'mv "{}" "{}"' | sed 's/\(.*\)02x/03x/'

Linux: comandi e shell

Comandi per la gestione di file e directoty	
cat nomefile	Visualizza il contenuto di un file
cp nomefile nomedirectory --parents	Duplica un file in un altro file mantiene la gerarchia delle directory
mv file1 file2	Rinomina il file
mv /lavoro/file1 /copia/file2	Sposta il file1 della directory lavoro nella directory copia rinominandolo in file2
rm – i *.dat	Cancella i file che hanno i caratteri "dat" dopo il punto, chiedendo conferma per ciascuna cancellazione
rm – r /lavoro	Cancella la directory lavoro con tutto il contenuto
ln file1 file2	Vengono assegnati più nomi allo stesso file
Locate prov*	Cerca in tutto il filesystem di Linux i file che iniziano con "prov"
umask	Stabilisce o modifica la maschera predefinita dei permessi per I file
Filtri sui file	
more nomefile	Visualizza il contenuto di un file 24 righe alla volta
sort nomefile	Ordina i dati dei file
sort +1 elenco	Ordina il file sul 2° campo
sort –n elenco	Ordina il file sul 1° campo in ordine numerico
sort –nr +2 elenco	Ordina il file sul 3° campo in ordine numerico decrescente
sort –u elenco	Ordina il file riducendo i duplicati
diff primofile secondofile	Confronta il contenuto di due file
diff –iw	elenco listaNella differenza vengono ignorate le differenza dovute a lettere maiuscole e minuscole
wc –l elenco	Conta le righe di elenco
wc –c elenco	in byte
grep opzioni espressione nomefile grep '[12]A' classi grep 'st.' elenco grep '\.xzw' elenco grep '^a' elenco grep 'o$' elenco	Ricerca una stringa all'interno di uno o più file Operazione di selezione *Opzioni* sono: **-i** : ignora la differenza tra minuscole e maiuscole **-h** : elimina la normale intestazione per la ricerca su più file **-n** : visualizza anche il numero di riga che contiene la stringa cercata **-l** : visualizza solo il nome dei file che contengono la stringa cercata **-L** : visualizza solo il nome dei file che NON contengono la stringa **-v** : visualizza le righe che non contengono la stringa cercata **-c** : visualizza il numero totale delle righe dei file che contengono la stringa cercata *Metacaratteri* usati da grep: **[]** racchiudono un insieme di caratteri, ciascuno dei quali può comparire in quella posizione **.** (punto) significa qualsiasi carattere \ toglie significato hai metacarreti ^ indica la ricerca a partire dal primo carattere di ogni riga **$** indica la ricerca a partire dalla fine della riga
cut opzioni nomefile	Sottrae parti delle righe di un file. Operazione di proiezione
cut –f2 elenco	Estrae dal file elenco le righe del secondo campo
cut –d ';' **–f**1 elenco	Estrae dal file elenco il primo campo, usando il punto e virgola come

	deliminatore dei campi
date | **cut –c**12-20 elenco	Estrae dall'output del comando date i caratteri dal 12 al 20
join primofile secondofile	Congiunge due file secondo valori uguali presenti nelle righe. Operazione di congiunzione
join Uno Tre	Effettua la congiunzione tra il primo e il terzo file sulla prima colonna uguale
join –j1 2 –j2 3 Uno Tre	Congiunge il file Uno e Tre usando il secondo campo (2) nel primo file (-j1) e il terzo campo (3) nel secondo file (-j2)
paste primofile secondofile	Unisce due file
head –n nomefile	Visualizza le righe iniziali di un file
head -2 elenco	Visualizza le prime 2 righe del file elenco
tail –/+n nomefile	Visualizza le righe a partire dalla fine del file
tail -12 elenco	Estrae le ultime 12 righe
tail +5 elenco	Estrae le righe a partire della quinta
Esempi di comandi	
ls –l | grep '^……..w'	Lista dei file che possono esser modificati da tutti gli utenti
wc –c *.txt	Visualizzare il numero di caratteri dei file con estensione .txt
| grep 'qualcosa' Utilizzati - E -o -h	Visualizza solo QUALCOSA, possibile utilizzare []{} look down Definisco un pattern Only matching Usato per restituire gli ip ma non il path All'interno di QUALCOSA devo delimitare con un / i caratteri fissi da proteggere
cut –d' ' –f2,5 elenco	Proiezione del 2° e 5° campo del file elenco usando come separatore dei campi lo spazio
cut –f1,3 persone | sort	Elenco alfabetico del file persone per il 1° e 3° campo
ls –l | cut –c34-42, 56-80 | sort +1	Elenco dei file presenti nella directory corrente, con la dimensione in byte, in ordine di nome
Comandi per la gestione del sistema e delle periferiche	
du	Fornisce informazioni sullo spazio occupato dai file. Un blocco è 1024 byte
tar azione pathname Le azioni possono essere: **-c** : usato per creare **-x** : usato per estrarre **-r** : aggiunge file a quelli già registrati **-t** : lista il contenuto dell'archivio **-v** : visualizza il nome dei file che vengono copiati (DA METTERE QUANDO VUOI VISUALIZZARLI) **-f** :da mettere a prescindere prima della dichiarazione del	Esempio : tar -cf (creo l'archivio) <files> (aggiungo i file all'archivio direttamente)

nome **-u** : crea backup solo se il file non è stato salvato in precedenza oppure se è stato modificato dopo l'ultimo backup. **-z** : specifica che si utilizza gzip Bisogna aggiungere .**gz** dopo .tar	tar -cvf <archive_name>.tar <files> tar -xvf <archive_name>.tar Path -name 's*.h' -exec tar -rvf file.tar {} \; Possibili utilizzi Tar... ....--exclude "path" (esclude i file di questo pat)....-cf nomearc.tar... ....-T - (aggiungiamo all'archivio i file trovati prima) (- ovvero usando come valore stdin).... .... find path -condizione -exec tar -rvf file.tar {} \; :Uso congiunto find e tar (mediante find ... -exec ...). Il parametro r di tar aggiunge (uno alla volta) I file trovati da find all'archivio senza ricrearlo.
ps	**stampa**
ps –ef	Visualizza le informazioni sullo stato dei processi attivi nel sistema (-e) e in modo completo)
kill pid	Provoca la terminazione del processo
kill -9 2345	Termina il processo n°2345 incondizionatamente (-9)
A > b	Override contenuto
A >> b	Aggiunge contenuto a poi b
| **awk** '{ s = s+$1 } END {print " Total used: ",s}'	Esegue la somma tra i valori della colonna 1 e stampa il risultato (total used: ...)
Awk -F 'separatore' '{print. $n} ' path	-F usato per printare solo un field (colonna) Separatore di ogni campo, per il path /etc/passwd è :
For user in $(....);do var=$(.....); echo "$user $var";done | sort	Quando serve avere due o piu variabili su una sola riga per eseguire un sort e non perdere nessuna di esse bisogna fare cosi
Find.....| du.,.....| awk.....	Per trovare la somma del peso dei file in un ramo
[0-9] {1,3}	Usato nel grep per dirgli di trovare un numero tra 0 e 9 che si ripete massimo 3 volte
for target in $(awk -F':' '{ print $1}' /etc/passwd); do res=$(find / -user $target -type f -printf '%p|%s\n' 2>/dev/null | sort -t | +1 -2 | awk -F| '{ s += $2 } END {if(s!=0){print "Sum: ",s}else{print "Sum: 0"}}'); [[$res != ""]] && echo "$target $res"; done | sort -k3 -n -r | head -n 3	Trova i tre utenti con la somma dei file piu grande senza aver problemi con gli utenti nulli
| **xargs ....comando** **|xargs -I{} comando {}**	Passa il risultato di ciò che sta prima della pipe al comando dopo uno alla volta Passa al comando l'imput ricevuto prima uno per volta
| **cat $file**	Passandogli con una variabile i file ne estrae il contenuto
sed -e 's/t/c && /' **-e 's/vs$/ns/'** **bash**	In base a **..** esegue un comando **c** quando trova il trigger **t** sulla vecchiastringa **vs** in questo caso sostituendola con la nuovastringa **ns** **S sta per script e dipende da -e** **.* come trigger per cambiare estensione, nel caso *. ....**

t sta per trigger, ciò che fa' partire la sostituzione	
c comando desiderato	
.. probabilmente sarà -e ovvero intera definisci una serie di comandi da eseguire	
Expr $(pipe) +- $(pipe2)	Espressione tra i valori della pipe1 e il valore della pipe2
REGULAR EXPRESSIONS **^A** **A$** **A^** **$A** **^^** **$$** **.** **^A$** **[0-9]** **"^A[a-z][a-z]"** **[^a-v]** **[a\b]**	 Cerca A all'inizio della linea Cerca A alla fine della linea Cerca A ovunque Cerca $A ovunque Cerca ^ all'inizio di una linea Cerca $ alla fine di una linea Cerca qualsiasi carattere Cerca una linea con solo A, si può sostituire con . per trovare una riga con qualsiasi carattere Cerca una linea con almeno un numero tra questi Per combinare piu regex, cerca una linea che inizia con A e due lettere minuscole Cerca lettere diverse da quelle dopo la ^ Cerca una linea con a o b
For $variabile in [lista]; do ..... ;done	Fa' un for
Strace 1/2/3>&1/2/3 Strace -e trace=memory ls 2>&1 | grep ')'	Restituisce le chiamate di un comando Reindirizzi stdinput (1) stoutput (2) stderror(3) in ..... Esempio di utilizzo su ls per far funzionare il grep
Find path condizione Find ..... -exec ..... {} /;	Trova le cose che corrispondono alla condizione nel path Alternativo utilizzo della pipeline
Password e utenti contenuti nel /etc/passwd	Utile per trovare gli utenti del computer
Assembly EBX,ECX,EDX Mov EAX, **OPCODE** Opcode 1 Opcode 4 _start: Comando_destinazione_partenza **Sys_exit:**	 Registri, utilizzabili per var Registro che assume diverse funzioni in base all'opcode Sys_exit---- EBX= cosa deve restituire (0=no error) Sys_write--- EBX=file su cui si vuole scrivere (1 per output) ECX=indirizzo var creata in .data EDX=lunghezza in byte della stringa Punto di entrata (deve essere definito in .text Global _start) **Mov ebx, 0 ; return0 status on exit -'no errors'** **Mov eax, 1; invoke SYS_EXIT IMPORTANTEEEE**

Interrupt da mettere dopo ogni syscall (EAX)	**Int 80h** **IMPORTANTEEEE**
Subroutines quando chiamate hanno un call e non un jump Quando ci chiede dove parte una subroutine la troviamo così, scrivendo la riga dove viene definita la sub, non ...	**Call (es)strlen** **IMPORTANTEEEE** **...chiamata**
Cicli Compara due registri Se ZF (indirizzo di eax) contiene 0 salta a ... Incrementa l'indirizzo puntato da eax	 Cmp Jz ... inc
Salta all'inizio del ciclo per ciclare	Jmp nome ciclo
Push il valore in ebx sullo stack per preservarlo mentre lo usiamo nella funzione	Push ebx
pop il valore sullo stack facendolo tornare dentro EBX	pop ebx
Ritorna dove la funzione viene chiamata	ret
PARTE DI C Pid=fork()...	Crea un processo figlio
...<0/==0/>0 (valori che la funzione sopra può restituire)	Abbiamo un errore/abbiamo il pid del figlio/abbiamo il ppid (padre)
Pid=wait(pid,&statloc,opts)	Aspetta che il processo figlio abbia terminato
S=execve(name, argv, envp)	Rimpiazza l'immagine del figlio con una nuova diversa dal padre
Exit (status)	Termina il processo in esecuzione e ritorna lo status
Pid=getpid()	Ritorna il pid di colui che lo chiama
Sleep()	Fa' dormire per tot secondi un processo,se processo figlio il padre termina, diventa così padre bash (1) se ivece si fa' sleep sul genitore il figlio quando termina diventa zombie perchè sta attendendo che il padre che dorme rilevi il suo stato di terminazione