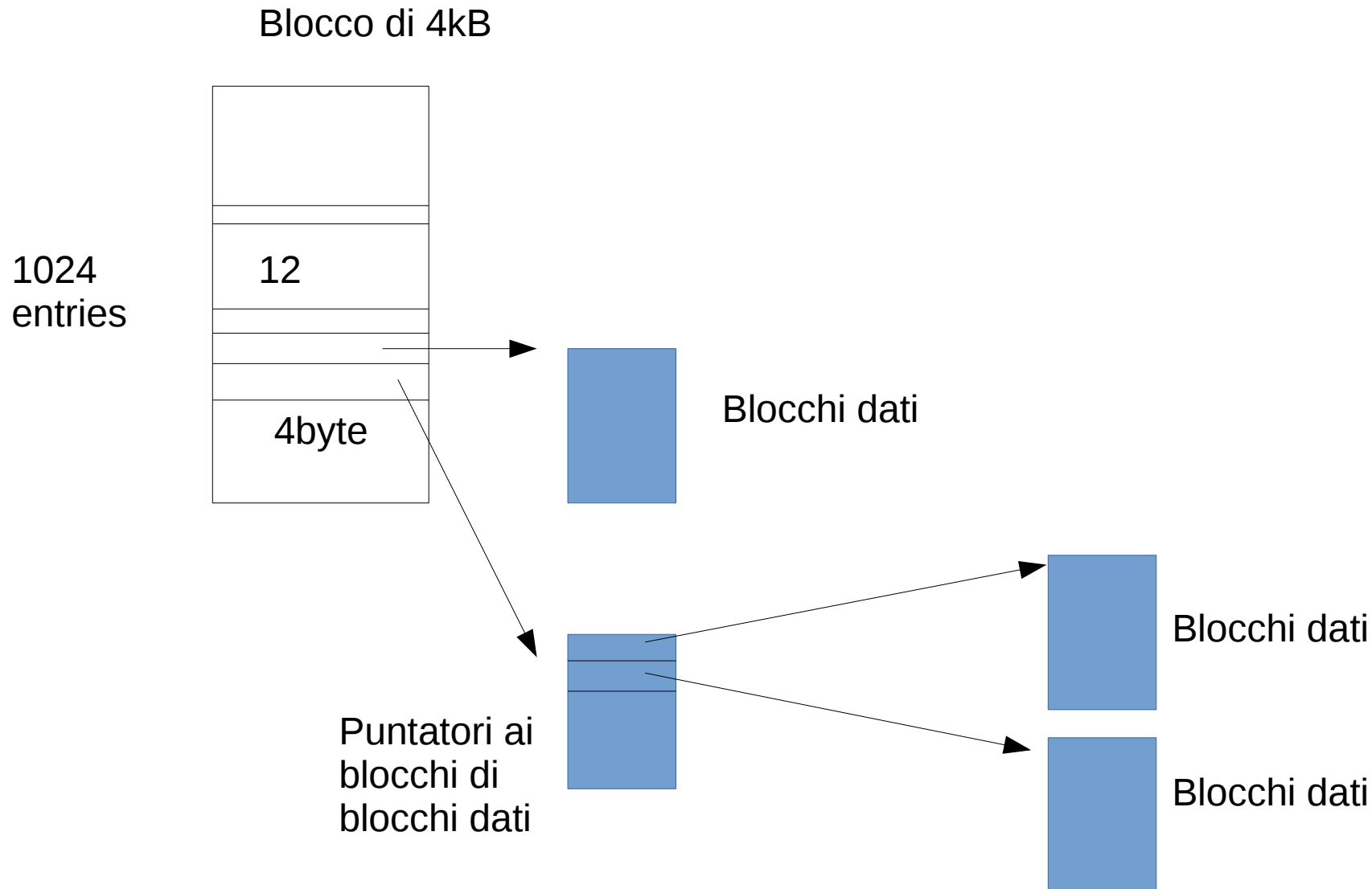


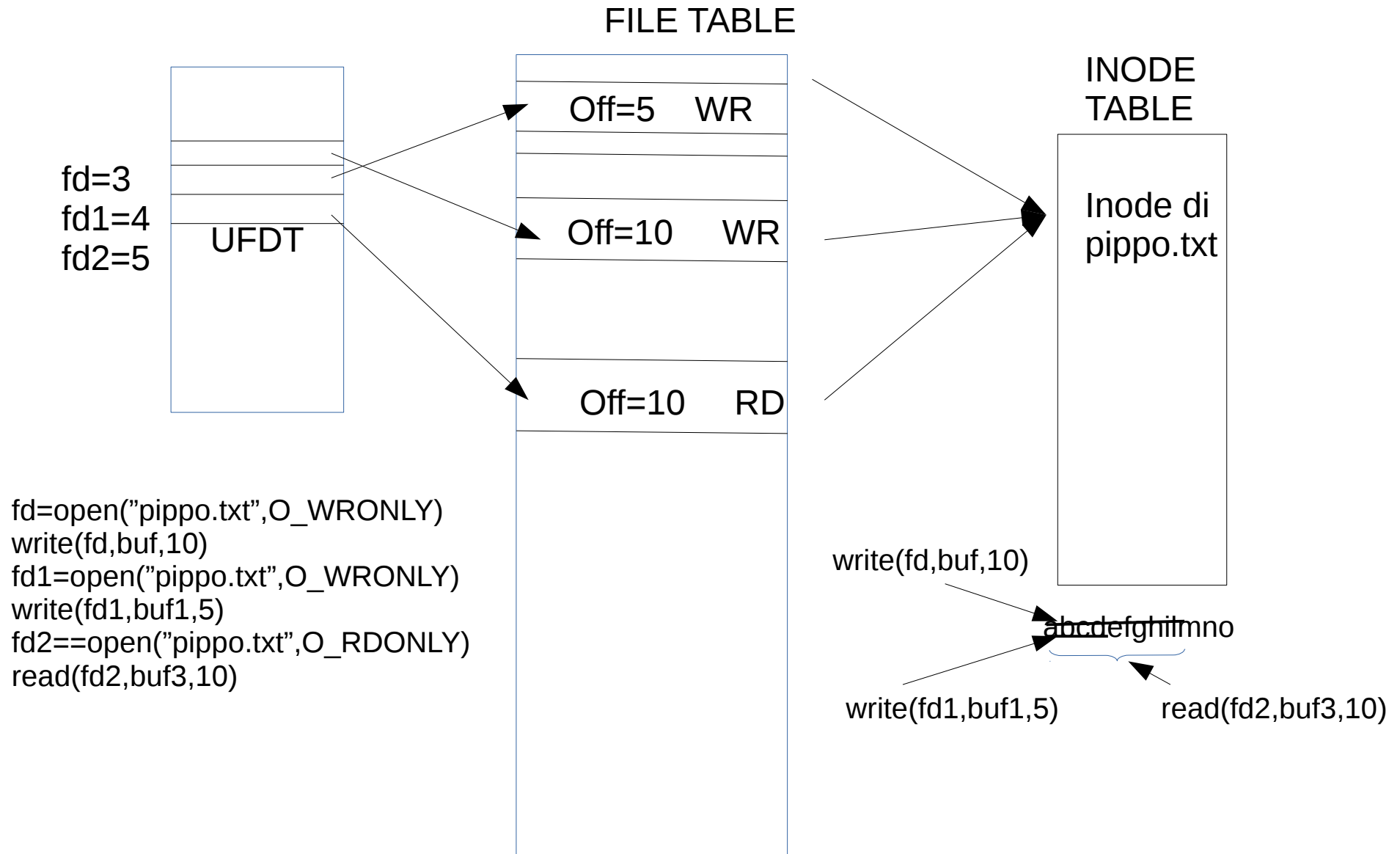
Qualche soluzione

Si supponga che un file system usi blocchi di 4 Kbyte, e che un puntatore ad un blocco occupi 4 byte. Se l'INODE di un file utilizza 12 blocchi, un blocco indiretto ed un blocco doppiamente indiretto, qual'è la massima dimensione dei file (in byte) ?

$$\text{size} = (12 + 1024 + 1024 * 1024) * 4\text{kB}$$



Si descrivono le tabelle utilizzate dal file system di Linux nel caso che un programma apra il file 'pippo.txt' tre volte in modalita' diverse (rispettivamente scrittura, scrittura e lettura) dopo che ogni apertura del file seguano rispettivamente una scrittura di 10 byte, una scrittura di 5 byte ed una lettura di 10 byte. Se il file contiene la stringa "abcdefghijklmno", come si modifica il file? Quali sono i 10 byte letti?



Scrivere uno script per scrivere le linee da n1 a n2 di un file dato con il simbolo di
redirezione senza usare tail/head

./Myhead n1 n2 < file

```
#!/bin/bash
n=0
While Read line
Do
  ((n=$n+1))
  If (( $n > $1 )) && ($n < $2 ))
  Then
    Echo $line
  Fi
Done
```

Scrivere uno script in Bash shell che lista tutti i file HTML di una directory. Per ogni file seleziona (se esiste) la riga <title> testo del titolo <\title> e scrive in un file l'informazione

...
file ...(nome file) → testo del
...

<title> stringa <\title>

```
for file in *.html  
do
```

```
  echo $file  
  while read line  
  do
```

```
    len=${#line} #lunghezza
```

```
    (( len = $len - 8 ))
```

```
    str=$(Echo $line|grep "<title>"|cut -c9-| cut -c1-$len)
```

```
    echo "$file → $str"
```

```
  done
```

```
done
```

stringa <\title>

stringa

```
./Extract > report.txt
```

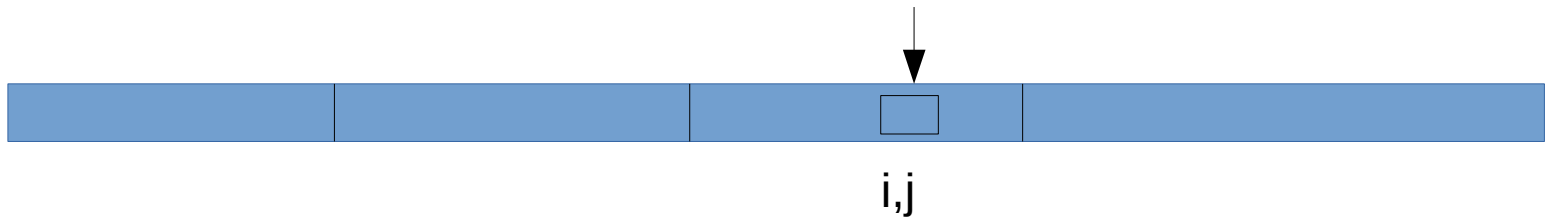
Esempio di nome dello script

Scrivere uno script in Bash per costruire una matrice di ordine N, con N dato come argomento in linea, inizializzare gli elementi con numeri random, e stampare la somma complessiva

```
./mat 5
```

```
i=0
while (( $i < $1 ))
do
  j=0
  while (( $j < $1 ))
  do
    (( n = $i*$1 + $j ))
    matrice[$n] = $RANDOM
  done
done
```

L'elemento di indice $(i*N + j)$ dell'array rappresenta l'elemento (i,j) della matrice



Contare tutti i link simbolici nella directory corrente con una riga di comando Linux

```
ls -l | while read l; do a=$(echo $l | cut -c 1-1); case $a in s) echo link; esac; done | wc -l
```

```
ls -l | cut -c1-1 | grep s | wc -l
```

Due soluzioni diverse: una complessa e una semplice

Scrivere un main in C che legge un file binario di 1000 numeri float e li memorizza in un array dinamico. Una funzione ne calcola il massimo ed il minimo che poi il main stampa.

```
#include
void funzione(float *f, int *a, int *b)
{
    *a=0; *b=1e6;
    for(i=0;i<1000,i++){
        if(f[i]>*a) *a=f[i];
        if(f[i]<*b) *b=f[i];
    }
}
main()
{
    float buf[1000];
    float *p;
    Int max , min;

    fd=open("file","O_RDONLY");
    read(fd,buf,1000*sizeof(float));
    p=(float*)malloc(1000*sizeof(float));

    for(i=0;i<1000;i++) p[i]=buf[i];
    funzione(p, &max,&min);
    printf(max, min)
}
```

Scrivere un main in C che crea ed inizializza in heap un array di 10 struct che rappresentano numeri complessi. Una funzione somma e visualizza i numeri.

Con puntatori

```
#include

typedef struct{
    float pr,pi;
} comp;

comp somma(comp*punt)
{
    typedef struct somma{float pr, pi;}
    somma spr;
    for(i=0;i<10;i++){
        spr.pr += punt[i].pr;
        spr.pi += punt[i].pi;
    }
    return spr;
}

main()
{
    comp *p, A;
    p=(comp *)malloc(10*sizeof(comp));
    A = somma(p);
    printf("somma=%f %f\n", A.pr, A.pi);
}
```

Senza puntatori

```
#include

typedef struct{
    float pr,pi;
} comp;

comp somma(comp*punt)
{
    typedef struct somma{float pr, pi;}
    somma spr;
    for(i=0;i<10;i++){
        spr.pr += punt[i].pr;
        spr.pi += punt[i].pi;
    }
    return spr;
}

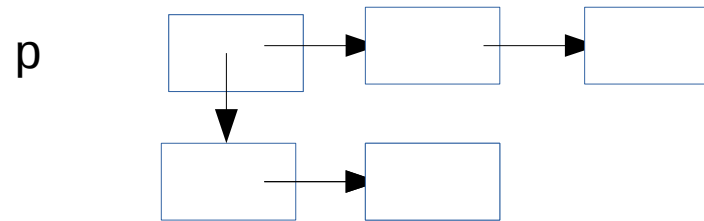
main()
{
    comp A[10], b;

    b = somma(&A[0]);
    printf("somma=%f %f\n", b.pr, b.pi);
}
```

Scrivere un main in C che legge con read() un file binario di interi in un array bidimensionale 5x4 allocato in heap. I dati sono memorizzati nel file per righe. Il main chiama una funzione che stampa gli elementi della matrice

1a riga	2a riga	5a riga
---------	---------	---------

```
#include
Main()
{
    Int fd, l, buf[20];
    fd=open("nome",O_RDONLY);
    read(fd,buf,4*5*sizeof(int));
    Int **p;
    p=malloc(5*sizeof(*int));
    for(i=0;i<5;i++){
        p[i]=malloc(4*sizeof(int));
        for(j=0;j<4;j++)  *(*p+i)+j)=buf[i*4+j]; /* *(*p+i)+j)=*(buf+i*4+j)
    }
    stampa(p);
}
stampa(int **punt)
{
    Int l,j;
    for(i=0;i<5;i++)
        for(j=0;j<4;j++)
            Printf("%d ", punt[i][j]);
}
```

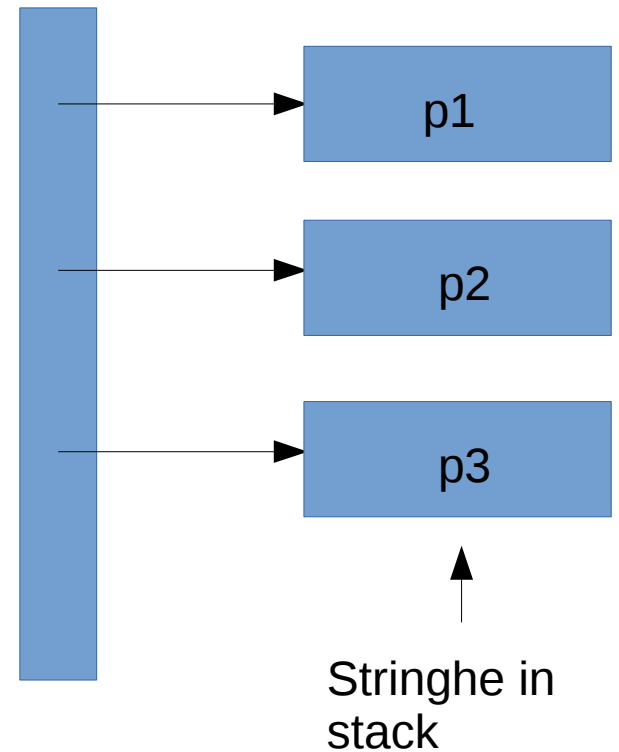


Per un array mono *p → p[i] o *(p+i) è uguale

Per un array 2D **p → p[i][j] o *(*p+i)+j) è uguale

Scrivere un main in C che crea in heap e inizializza a piacere un array bidimensionale 3x10 di stringhe. Una funzione scrive per righe l'array su un file

```
#include
main()
{
    char **p, *p1, *p2,*p3;
    p1="sdfsaf"; p2="rrrre";p3="gsdfg";
    p=malloc(3*sizeof(*char));
    p[0]=p1;
    p[1]=p2;
    p[3]=p3;
    stampa(p);
}
stampa(int **punt)
{
    Int l,j;
    for(i=0;i<3;i++)
        for(j=0;j<10;j++)
            printf("%s ", punt[i][j]);
}
```



Array di
puntatori in
heap

Inizializzare matrici di stringhe in heap

```
#include
Main()
{
    char **p, *p1, *p2, *p3;

    p1=malloc(10); p2=malloc(10);p3=malloc(10);
    p1="sdfsaf"; p2="rrrre";p3="gsdfg";
    p=malloc(3*sizeof(*char));
    p[0]=p1;
    p[1]=p2;
    p[3]=p3;
    stampa(p);
}
stampa(int **punt)
{
    int i,j;
    for(i=0;i<3;i++)
        printf("%s ", punt[i]);
}
```

Chiamiamo RUID, EUID “Real User ID” ed “Effective User ID”.

Si supponga che una directory di Linux contenga i seguenti file, dove l'unico file eseguibile è all:

Filename Owner Group SetUID Owner Group Other

a10.txt	15	15	s	rw-	r--	---
b10.txt	15	99	-	---	rw-	rw-
c12.txt	75	75	s	rw-	rw-	---
c14.txt	25	25	-	rw-	r--	r--
dati.txt	25	99	s	--x	r-x	---
all	75	75	s	--x	--x	--x

Si supponga inoltre che l'utente 15 sia nel gruppo 99 e che l'utente 25 sia nel gruppo 25.

Rispondere alle seguenti domande:

a. Quali file può leggere un processo con RUID=15, EUID=15, EGID=15?

b. Quali file può scrivere un processo con RUID=15, EUID=25, EGID=75?

c. Quali file può leggere un utente con UID=100 GID=100 che esegue il programma all?

- a Può leggere a10.txt c14.txt
- b Può leggere b10.txt c12.txt c14.txt
- c Può leggere b10.txt c12.txt c14.txt

Il file lista contiene il codice eseguibile di un programma che lista il contenuto di file testuali (l'uso del comando è quindi del tipo: \$lista <file>). Il file lista è dell'utente Giulio, con UID=1500 e GID=2000, ed ha le protezioni 510. Per vedere le caratteristiche del file lista, l'utente Giulio effettua il comando: \$ls -l > dati

Facendo \$ls -l dati Giulio vede che il file dati è ovviamente di Giulio ed ha le seguenti protezioni: 604.

- a) Può Giulio vedere il contenuto del file dati con il comando \$lista dati ? **Sì**
- b) Cosa succede se Giorgio del gruppo 2000 scrive il comando \$lista dati? **Può eseguire lista ma non può leggere dati**
- c) Cosa succede se Giorgio del gruppo 2000 scrive il comando \$lista dati dopo che Giulio setta il bit SetUID del file lista?
Giorgio può eseguire lista e può leggere dati

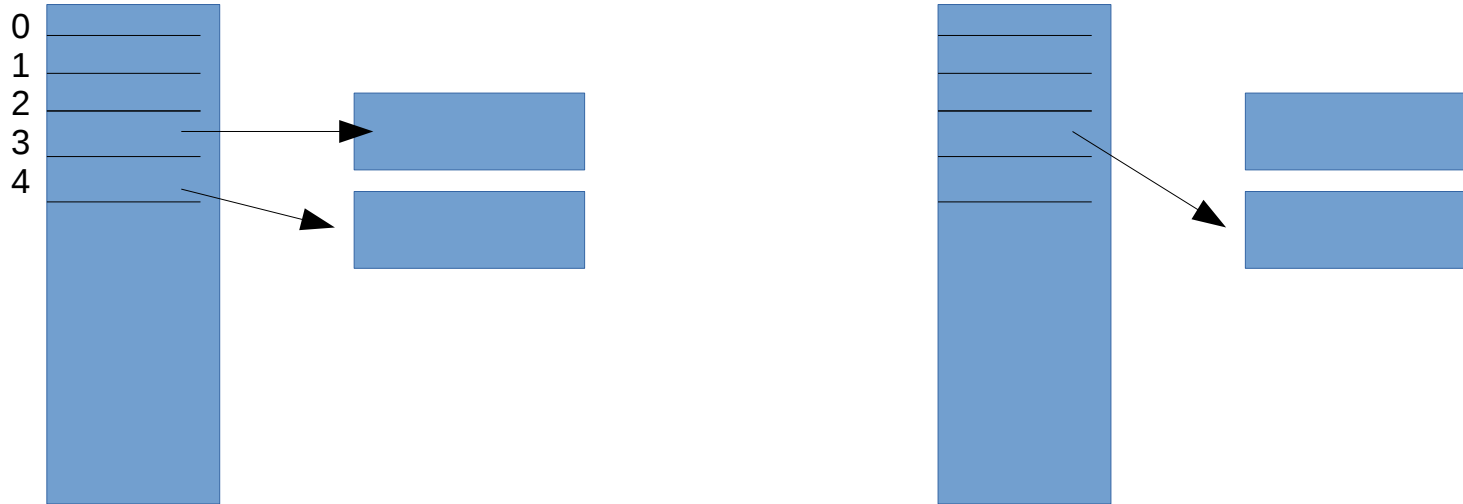
File eseguibile lista: r-x -x ---

Dati: rw- --- r--

dopo che Giulio setta il bit: File eseguibile lista s r-x -x ---

Quando Giorgio esegue \$lista dati diventa il proprietario di dati

Scrivere un main in C che apre due file di 1000 byte in lettura e scrittura. Legge il primo file e usando lo stesso descrittore copia il contenuto nell'altro file usando le SC read(), write() e dup2(source,dest).



```
void main(int argc, char *argv[])
{
    fd1=open(argv[1],O_RDWR);
    fd2=open(argv[2],O_RDWR);

    read(fd1,buf,1000);
    dup2(4,3);
    write(fd1,buf,1000);
}
```