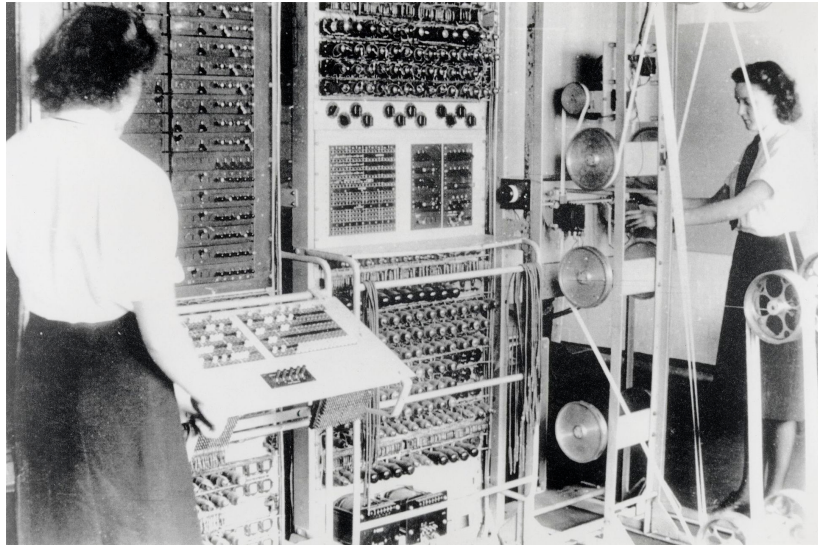


# Programadoras dos ordenadores Colossus (1943)



- Primeiras calculadoras electrónicas
- Programadas por 273 mulleres programadoras do Women's Royal Naval Service
- Entre outras (2016): Irene Dixon, Lorna Cockayne, Shirley Wheeldon, Joanna Chorley and Margaret Mortimer
- Usados por Inglaterra para descifrar comunicacións alemáneas na 2ª guerra mundial



# Arquivos

## Lectura de datos dende un arquivo:

- Comando **load**: lectura de arquivos numéricos (sen letras nin símbolos) onde tódalas liñas teñen o mesmo número de elementos
- **Non** necesita abrir e pechar o arquivo
- Exemplos:

*load datos.dat* => carga os datos á matriz *datos*

*x=load('datos.dat');* => carga os datos á matriz *x*

Mellor a segunda forma con nomes de arquivos longos.

# Entrada e saída a arquivos

- Apertura:  $f=fopen('arquivo.dat', 'permisos');$
- Retorna  $f=-1$  en caso de erro,  $f>0$  noutro caso.
- Permisos:
  - 'r': abre o arquivo para lectura (por defecto)
  - 'w': escritura: borra o arquivo se xa existe
  - 'a': abre o arquivo para escribir ao final (conserva o que xa está)

```
nf='arquivo.txt';f=fopen(nf,'r');  
if -1==f; error('fopen %s',nf); end
```

- Peche do arquivo:  $fclose(f);$

# Escritura / lectura en archivos

- **Escribir:** función *fprintf*:

*fprintf(f, 'formato', datos);*

- A mesma función para saída por pantalla pero con *f* para enviar a arquivo. Ex: *fprintf(f, 'n=%i x=%f\n', n, x);*
  - Se mostra en pantalla, non retorna nada; se almacena en arquivo, retorna o nº de bytes escritos (rematar en ;)
  - Vectorizada: *x=randi(100,1,20); fprintf(f, '%i ', x);*
- **Ler:** Función *fscanf*: ten dúas formas:

*a=fscanf(f, 'formato');*  
*[a m]=fscanf(f, 'formato', n);*

Le *n* datos. Só con arquivos nos que as liñas teñen distintos números de elementos ou conteñen texto (noutro caso, usa *load*)

Se non lle indicas *n*, le tódolos datos do arquivo

# Lectura de un archivo con *fscanf*

$$[a,m] = fscanf(f, 'formato', n)$$

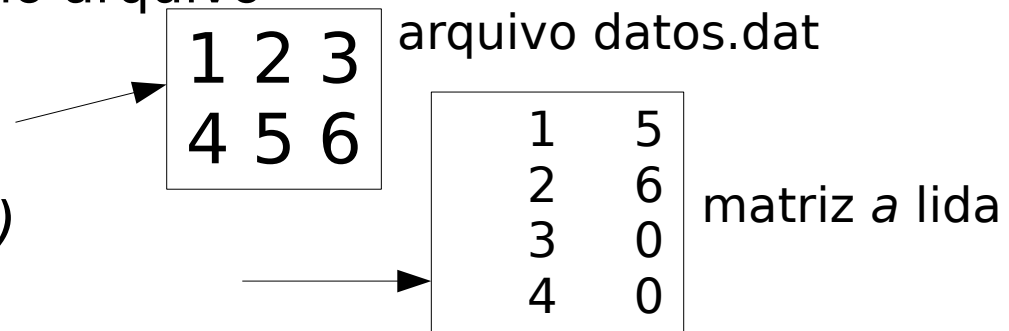
- *f*: identificador de archivo retornado por *fopen*
- *'formato'*: cadeia de formato: igual que en *fprintf*
- Se non ten nada que ler (p. ex., no final do arquivo) retorna *a=[]*
- *m*: nº de datos realmente lidos
- Argumento *n* (opcional, se non está vale *inf*): nº de datos a ler: pode ser:
  - Un enteiro: neste caso, le *n* datos (ou  $m < n$  se non hai máis datos no arquivo), que se almacenan no vector columna *a*
  - *inf*: Le tódolos datos do arquivo e almacénaos no vector columna *a* (igual que se non se especifica *n*).
  - *[nf nc]*: Le *nf* x *nc* datos do arquivo e méteos por columnas nunha matriz *a* de orde *nf* x *nc*.

# Lectura desde un archivo con *fscanf*

- Se hai menos de  $nf \times nc$  datos no arquivo, le os que haxa e enche os elementos restantes con ceros
- O valor  $nc$  pode ser *inf*, de modo que  $a$  ten  $nf$  filas e o número mínimo de columnas para mete-los datos lidos en  $a$
- O valor  $nf$  non pode ser *inf*
- Se  $n=[nf \ nc]$ , entón  $m$  pode ser menor que  $nf \times nc$  cando hai menos de  $nf \times nc$  datos no arquivo

- Exemplo:

```
f=fopen('datos.dat');  
a=fscanf(f,'%d', [4 inf])  
fclose(f);
```



- A matriz  $a$  é  $4 \times 2$  porque en *datos.dat* hai 6 elementos e non collen nunha matriz  $4 \times 1$ , necesita 2 columnas

# Exemplos de lectura/escritura

- Lectura dunha matriz  $[nf \times nc]$  dende un arquivo:  
***a = load('arquivo.txt');***

```
f = fopen('arquivo.txt', 'r');  
a = fscanf(f, '%g', [nf nc]);  
fclose(f);
```

- Escritura nun arquivo:

```
f = fopen('arquivo.txt', 'w');  
for i=1:nf  
    fprintf(f, '%g ', a(i,:));  
    fprintf(f, '\n');  
end  
fclose(f);
```

- Lectura dun arquivo irregular completo:

```
1 2 3  
4 5  
6 7 8 9 8 7 6
```

```
f = fopen('arquivo.txt', 'r');  
x = fscanf(f, '%g');  
fclose(f);
```

- Adición ao final dun arquivo:

```
x = [1 2 3 4];  
f = fopen('arquivo.txt', 'a');  
fprintf(f, '%g ', x);  
fclose(f);
```

# Lectura de cadeas de caracteres con *fgetl* e *strsplit*

- Sintaxe: *s=fgetl(f)*; le unha liña como cadea de caracteres
- Se a liña está baleira, ou está ao final do arquivo, retorna -1 como número
- Divides en palabras con *strsplit(s)*: *s{1},...,s{n}*
- Se a liña ten números, hai que dividir a cadea en palabras e converter os números en double (función *str2double*)
- Se a cadea de caracteres non é un número, *str2double* retorna *NaN*, e podes comprobalo coa función *isnan*

```
clear all
f=fopen('datos.dat');
if -1==f; error('fopen datos.dat'); end
while ~feof(f)
    s=fgetl(f);
    fprintf('liña= <%s>\n',s);
    t=strsplit(s);n=numel(t);
    fprintf('palabras:');
    for j=1:n
        x=str2double(t{j});
        if ~isnan(x)
            fprintf('%f\n',x);
        else
            fprintf('%s\n',t{j});
        end
    end
end
fclose(f);
```

Divide a cadea e converte a números se procede



# Funcións *feof* (fin de arquivo) e *frewind* (rebobinado)

- Cando lemos, é importante saber cando chegamos á fin de arquivo
- A función *fscanf* retorna 0 bytes cando chega á fin do arquivo
- A función *feof(f)* retorna 1 se atopou a fin do arquivo, ou 0 en caso contrario
- Función *frewind(f)*: retorna ao comezo do arquivo

Exemplo: programa que le todo o arquivo e mostra liña a liña:

```
f=fopen('datos.dat');  
if -1==f  
    error('fopen datos.dat')  
end  
while ~feof(f)  
    s=fgetl(f);fprintf('%s\n',s);  
end  
fclose(f);
```

# Funcións *fseek*, *ftell*, *str2double*, *isnan*

- *fseek(f,n,w)*: sitúase  $n$  bytes (caracteres) á dereita (se  $n > 0$ ) ou esquerda (se  $n < 0$ ) de  $w$ , que pode ser:
  - ‘bof’ ou -1: comezo do arquivo (*beginning of file*)
  - ‘cof’ ou 0: posición actual no arquivo (*current position on file*)
  - ‘eof’ ou 1: final do arquivo (*end of file*)
- *fseek(f,10,'bof')*: 10 caracteres logo do comezo do arquivo
- *fseek(f,-5,'cof')*: 5 caracteres antes da posición actual
- *fseek(f,-20,'eof')*: 20 caracteres antes do final de arquivo
- *ftell(f)*: retorna a posición actual (onde se vai ler ou escribir) en caracteres dende o comezo do arquivo
- Distinguir se  $s$  é número ou palabra:  $x = \text{str2double}(s)$ ,  $\text{isnan}(x) = 1$  se  $x$  é palabra,  $= 0$  se  $x$  é número (real ou enteiro).

# Exemplo de lectura de arquivo en formato *write.table* de R con *fscanf*

- Arquivo co seguinte contido:

```
clear all
f=fopen('arquivo.dat','r');
if -1==f
    error('erro abrindo arquivo.dat')
end
```

|   | E1    | E2    | E3   | E4    | Saida  |
|---|-------|-------|------|-------|--------|
| 1 | 0.25  | 0.33  | 1.23 | -0.51 | Branco |
| 2 | -0.34 | 1.3E5 | 0.22 | 4.3   | Negro  |

```
s=strsplit(fgetl(f));s(1)=[];nc=numel(s)-1;nf=0;
while ~feof(f)
    fgetl(f);nf=nf+1;
end
```

Le o nº de filas  
e columnas

```
dato=zeros(nf,nc);saida=cell(1,nf);
frewind(f);nomes=fscanf(f,'%s',nc+1);
for i=1:nf
    fscanf(f,'%i',1); % le e descarta (non almacena) o nº de liña
    dato(i,:)=fscanf(f,'%g',nc); % le as ne entradas (tamén formato exponencial)
    saida{i}=fscanf(f,'%s',1); % le a saída (cadea de caracteres)
end
fclose(f);
for i=1:nf
    fprintf('dato %i: ',i);fprintf('%g ',dato(i,:));fprintf('saida=%s\n',saida{i});
end
```