

Computación 1



Manejo de archivos en Octave

Introducción



- Necesidad de persistencia de datos:
 - Resultados numéricos
 - Base de datos
- Funcionalidad para cargar o recuperar datos desde archivos.

Comandos utilitarios



Cuando estamos trabajando con archivos hay varios comandos de Octave que conviene conocer.

- **type:** permite mostrar el contenido de un archivo de texto en la consola.
- **pwd:** muestra en pantalla la ruta del directorio de trabajo donde estamos posicionados.
- **dir:** muestra el contenido
- **cd:** cambia de directorio

Los comandos **load** y **save**



- Podemos guardar variables en archivos de texto mediante el comando Octave **save**.
- Las variables guardadas las podremos recuperar luego mediante el comando **load**.

El comando save



- **save [nombre_archivo]**: Guarda todas las variables del workspace en un archivo.
- **save [nombre_archivo] v1 v2**: Se pueden especificar qué variables concretas queremos guardar en un archivo.
- **save options [nombre_archivo] v1 v2**: donde **options** son una o más opciones separadas por espacio. Las opciones nos permiten especificar, por ejemplo, que se guarden los archivos en distintos formatos binarios o que se guarden los datos en formato comprimido zip.

Lectura y escritura de archivos de texto



- La función **fopen()** y **fclose()**
 - Para poder realizar operaciones de lectura y/o escritura en los archivos de texto es necesario primero abrir el archivo con **fopen()**.
 - Una vez abierto el archivo, se procede a realizar las distintas operaciones de lectura y/o de escritura en el mismo.
 - Cuando se termina de operar con el archivo hay que cerrarlo con la función **fclose()**.

Lectura y escritura de archivos de texto



El procedimiento habitual para leer datos de un archivo se resume en el siguiente esquema:

```
file = fopen('miarchivo.txt', 'r');  
% operaciones de lectura del archivo  
fclose(file);
```

Lectura y escritura de archivos de texto(cont.)



- **file = fopen(filename, permission)**
 - **filename**: cadena de texto con el nombre del archivo que se quiere abrir. Si incluye la ruta del archivo se utilizará, si no se buscará o creará el archivo en el directorio de trabajo.
 - **permission**: especifica el modo de apertura del archivo. *permission* es una cadena de texto que puede tener los siguientes valores:
 - ‘**r**’: Abre el archivo para lectura. Es el modo por defecto, si se utiliza la función `fopen()` sin el argumento *permission*
 - ‘**r+**’: Abre el archivo en modo lectura-escritura
 - ‘**w**’: Abre o crea un nuevo archivo en modo escritura. Si existe se sobrescribe

Lectura y escritura de archivos de texto(II)



- **file = fopen(filename, permission)**
 - **‘w+’**: Abre o crea un archivo para lectura-escritura. Si existe, se sobrescribe
 - **‘a’**: Abre o crea un nuevo archivo para escritura. Si existe el archivo, añade al final del mismo.
 - **‘a+’**: Abre o crea un nuevo archivo para lectura-escritura. Si existe, se añade al final del mismo.

Escritura de datos formateados



fprintf()

- Ejemplo:

```
A = [11 12 13 14; 21 22 23 24; 31 32 36 34];
```

```
file = fopen ("prueba.txt", "w");
```

```
fprintf (file, '%d ', A)
```

```
fclose(file)
```

```
prueba.txt x  
1 11 21 31 12 22 32 13 23 36 14 24 34
```

Escritura de datos formateados(cont.)



fprintf()

- Otro ejemplo:

```
A = [11 12 13 14; 21 22 23 24; 31 32 36 34];
```

```
file = fopen ("prueba.txt", "w");
```

```
fprintf (file, '%d %d %d %d\n', A)
```

```
fclose(file)
```

```
prueba.txt x
1 11 21 31 12
2 22 32 13 23
3 36 14 24 34
```



fscanf()

La lectura de datos a partir de un archivo se realiza mediante los comandos:

- `[A,cont]=fscanf(fid,'formato')`
- `[A,cont]=fscanf(fid,'formato',size)`

Lectura de datos(cont.)



Ejemplo:

Se supone que en la carpeta de trabajo de Octave se encuentra un archivo de nombre `datos.txt`, cuyo contenido es:

1 2 3 4 5

6 7 8 9 10

Lectura de datos(cont.)



```
>>fid=fopen('datos.txt','r')
```

```
>>[A,cont]=fscanf(fid,'%d')
```

La salida es el vector columna de contenido:1 2 3 4 5 6 7 8 9 10 y cont=10.

Otra posibilidad:

```
>>[A,cont]=fscanf(fid,'%d',[2,5])
```

```
A=
```

```
1 3 5 7 9
```

```
2 4 6 8 10
```

```
cont=10
```

Lectura de archivos línea a línea



Una vez abierto el archivo para lectura podemos ir leyendo una línea cada vez, que recibiremos como una cadena de texto, con o sin carácter fin de línea, según la función de lectura utilizada:

- **line = fgets(file):** Lee una línea del archivo **file**, incluyendo el carácter de fin de línea, y la devuelve en forma de cadena de texto.
- **line = fgets(file, nchar):** Lee al menos **nchar** caracteres de la siguiente línea del archivo **file**. (Si se alcanza el final de línea o el final del archivo devuelve lo que haya leído hasta ahí).
- **line = fgetl(file):** Lee la siguiente línea del archivo **file**, sin incluir el carácter fin de línea, y la devuelve como una cadena de caracteres.