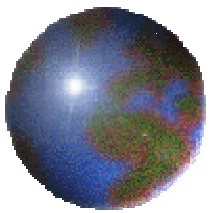


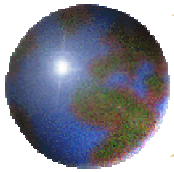
Desarrollo de Portales WEB
Lenguaje PHP

INSTITUTO SUPERIOR TECNOLÓGICO
IDAT
R.M. N° 420 - 94 - ED

Gonzalo Anchante Hurtado

Docente Carrera Profesional

Computación e Informática



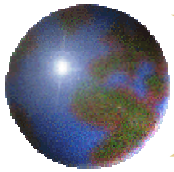
Lectura selectiva: fgets y fread

- Si queremos especificar la cantidad de archivo que queremos recuperar podemos usar fgets o fread

Lectura selectiva: fgets

- sintaxis fgets (file_handler, longitud)

- lee el contenido de un archivo (abierto con fopen) hasta la longitud indicada (- 1 byte), o hasta el primer fin de línea, o hasta el fin del archivo (es decir, hasta que se de la primera de esas tres condiciones).



Lectura selectiva: fgets

- Una función usada en conjunción con fgets es feof(), que chequea si hemos llegado o no al final del archivo.

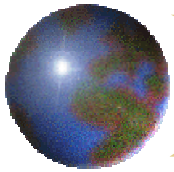
Lectura selectiva: fgets

- Función similar a fgets() es fgetss, con la única diferencia de que si se trata de un archivo html, elimina las etiquetas html o php que puedan haber en la porción recuperada. Admite el parámetro allowable_tags donde puedes incluir aquellas etiquetas que no deseas que sean removidas.

Lectura selectiva: fread

- sintaxis fread (file_handler, longitud)

- muy similar a fgets. La principal diferencia es que fread lee los archivos en modo binario.

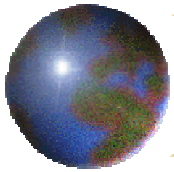


Lectura selectiva: fread

- Conjuntamente podemos usar filesize() (solo con archivos locales), para obtener el tamaño del archivo y pasarle el valor como longitud a fread.

Lectura selectiva: fread

- Ejercicio 4

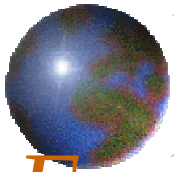


Escribiendo contenido: fwrite, fputs

- fwrite y fputs son funciones idénticas. Ambas nos permiten escribir una línea nueva en el archivo (abierto con fopen). La escritura tendrá lugar sobrescribiendo el contenido o añadiéndolo al final, según el modo usado con fopen.

Escribiendo contenido: fwrite, fputs

- Sintaxis:
  - fwrite (file_handler, texto_a_escribir [, longitud])
  - fputs (file_handler, texto_a_escribir [, longitud])



Escribiendo contenido: fwrite, fputs

- El parámetro opcional "longitud" nos permite especificar la longitud de la cadena a escribir.
- Si no lo empleamos, se escribirá entera.
- Si lo empleamos y la cadena es mas larga que la "longitud", solo se escribirá el numero de caracteres permitido por esta.

Escribiendo contenido: fwrite, fputs

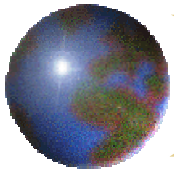
- La función retorna un numero entero, que será -1 si falla la operación, o el numero de caracteres escritos.
- Estas funciones no incluyen saltos de linea, por tanto debes añadirlos si los deseas: '\n' en linux, **'\r\n'** en Windows.

Escribiendo contenido: fwrite, fputs

- Ejercicio 5

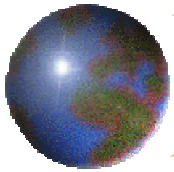
Conflictos escritura/lectura

- Nuestras páginas php serán visitadas (sin duda) por un gran número de personas; es posible que en algún momento dos o mas personas soliciten el mismo archivo a la vez. Mientras esto no es ningún problema cuando se trata de archivos de solo lectura, si puede llegar a serlo, y grave, cuando se trate de archivos con permisos de lectura/escritura



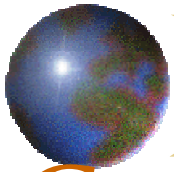
Conflictos escritura/lectura

- Pensemos en el ejemplo típico, un contador de visitas que a cada nuevo visitante abre el archivo donde se guardan los datos, pone dentro la información de la nueva visita, y después lo cierra. ¿que pasaría si durante la operación de escritura hay otro usuario que visita la web? que el script abriría de nuevo el fichero y escribiría los datos antes de que la anterior operación hubiera terminado.



Conflictos escritura/lectura

- Si coinciden los dos procesos, como mal menor, la información se escribirá de forma incorrecta (y por tanto inútil). Como mal peor, tendremos un archivo corrupto.



Conflictos escritura/lectura

- Para evitar estos problemas al manejar archivos, con php disponemos de la **función flock (file lock)** con la cual podemos poner un *candado* al archivo (impidiendo que se abra en modo escritura) o reabrirlo a voluntad.

Conflictos escritura/lectura = flock

flock() opera sobre un file handler que debe apuntar a un archivo previamente *abierto*. Puede tener los siguientes valores:

- Para que adquiera un bloqueo compartido (lectura), se fija su valor a 1. Mas de un proceso puede tener un candado compartido para un mismo archivo.
- Para obtener un bloqueo exclusivo (escritura), se fija el valor a 2. Solo un proceso puede tener un candado exclusivo. Los demás procesos deben esperar
- Para liberar un bloqueo (compartido o exclusivo), se fija el valor a 3.
- Usando esta función pueden implementarse mecanismos de sincronización entre procesos