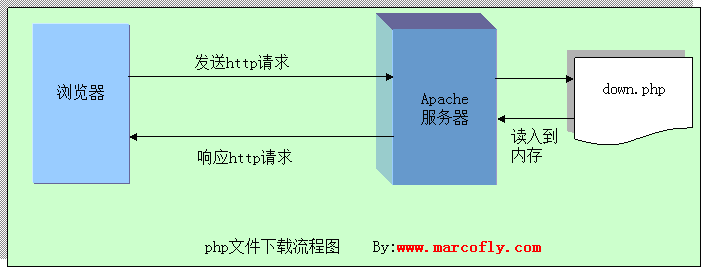
客户端从服务端下载文件的流程分析：   
  
浏览器发送一个请求，请求访问服务器中的某个网页（如：down.php），该网页的代码如下。   
服务器接受到该请求以后，马上运行该down.php文件   
运行该文件的时候，必然要把将要被下载的文件读入内存当中（这里是圣诞狂欢.jpg这张图片），这里通过fopen()函数完成该动作   
注意：任何有关从服务器下载的文件操作，必然需要先在服务端将文件读入内存当中   
  
现在文件已经在内存当中了，这是需要从内存当中读取文件，通过fread()函数完成该动作   
需要注意的是，如果文件较大，文件应该是被分成多段返回给客户端的，并不是等文件在服务端全部读取完毕后，一次性返回给客户端，因为这样子会增加服务器的负荷。   
所以我们需要在php代码中设置一次读取的字节数,比如我在下面的代码中通过$buffer=1024设置一次读取的字节数，每读取一次，就输出数据（即返回给浏览器）



<?php   
header("Content-type:text/html;charset=utf-8");   
// $file\_name="cookie.jpg";   
$file\_name="圣诞狂欢.jpg";   
//用以解决中文不能显示出来的问题   
$file\_name=iconv("utf-8","gb2312",$file\_name);   
$file\_sub\_path=$\_SERVER['DOCUMENT\_ROOT']."marcofly/phpstudy/down/down/";   
$file\_path=$file\_sub\_path.$file\_name;   
//首先要判断给定的文件存在与否   
if(!file\_exists($file\_path)){   
echo "没有该文件文件";   
return ;   
}   
$fp=fopen($file\_path,"r");   
$file\_size=filesize($file\_path);   
//下载文件需要用到的头   
Header("Content-type: application/octet-stream");   
Header("Accept-Ranges: bytes");   
Header("Accept-Length:".$file\_size);   
Header("Content-Disposition: attachment; filename=".$file\_name);   
$buffer=1024;   
$file\_count=0;   
//向浏览器返回数据   
while(!feof($fp) && $file\_count<$file\_size){   
$file\_con=fread($fp,$buffer);   
$file\_count+=$buffer;   
echo $file\_con;   
}   
fclose($fp);   
?>

几点注意事项：   
  
header("Content-type:text/html;charset=utf-8")的作用：在服务器响应浏览器的请求时，告诉浏览器以编码格式为UTF-8的编码显示该内容   
关于file\_exists()函数不支持中文路径的问题:因为php函数比较早，不支持中文，所以如果被下载的文件名是中文的话，需要对其进行字符编码转换，否则file\_exists()函数不能识别，可以使用iconv()函数进行编码转换   
$file\_sub\_path() 我使用的是绝对路径，执行效率要比相对路径高   
Header("Content-type: application/octet-stream")的作用：通过这句代码客户端浏览器就能知道服务端返回的文件形式   
Header("Accept-Ranges: bytes")的作用：告诉客户端浏览器返回的文件大小是按照字节进行计算的   
Header("Accept-Length:".$file\_size)的作用：告诉浏览器返回的文件大小   
Header("Content-Disposition: attachment; filename=".$file\_name)的作用:告诉浏览器返回的文件的名称   
以上四个Header()是必需的   
fclose($fp)可以把缓冲区内最后剩余的数据输出到磁盘文件中，并释放文件指针和有关的缓冲区