## copy file Using low-level I/O

		•				
	0 .				n-1	
•				MARTINEZ-MANORIZERA POR POPURENA POR ARTONIA MARTINEZA MARTINEZA POR PORTUGUEN		Source
	principal de la principal de l				a chunk of	hules
	•		e e	•	V S S S S S S S S S S S S S S S S S S S	oy res
1	each ti	me we	like to	read	n BUFFER_S	IE (1024)
	bytes a	nd write	into the	destir	nation File /	
	IIID VP	peat th	is proces	s until	all chunks	are read;
(2)	This	is perform	ned by -	the our	ter while Loc	·p /
3	The re	eason to c	opy in	biece-wi	se fashion is	that
	we d	lo not know	u the s	ize of.	the Source File	z, we
	can r	not read	in the	uhole	File into me	mory,
	Especi	ally when	the siz	e of or	riginal File is	larger
	than-	the RAM	1 size	1		
					while Loop,	1.

For each iteration of the outer while Loop,
it read in one chunk of data from source File.

BUFFRE\_SIZE

Source

Source

Source

Actual size of the data read in by read;

could be smaller than the expected chunk size

1024;
Actual size specified by buffer chars

The inner while Loop writes the data in buffer (size is buffer Chars) into the destination File.



- bp points to the Start position of the data you write back
- bp is moving toward the end of buffer array !
- Because the writer, function may not write the same amount of bytes that we specified in the parameter. E.g. writer ufd, bp, buffer chars) may not write back "buffer chars' characters into file.

It could be less than that amount!

- That is why we need an inner Loop for writes! We make sure all data that has been stored in buffer is completely write back to destination File!
- bufferchars is decreased after each write () call, meaning the remaining bytes of data that has Not write back yet!
- bp (the pointer) is increased after each write, meaning the start position for the next write operation
- We continue the inner Loop until remaining data is Zero bytes; (buffer Chars == 0)!