**Title:** Redis SETBIT

***Excerpt:*** The SETBIT command is used to set a bit value to 0 or 1 at a given offset. This command returns the original value of the bit upon calling the command. The most important fact about the SETBIT is that it operates on O(1) linear time complexity.

**Permalink:** redis-setbit

**Category:** Redis

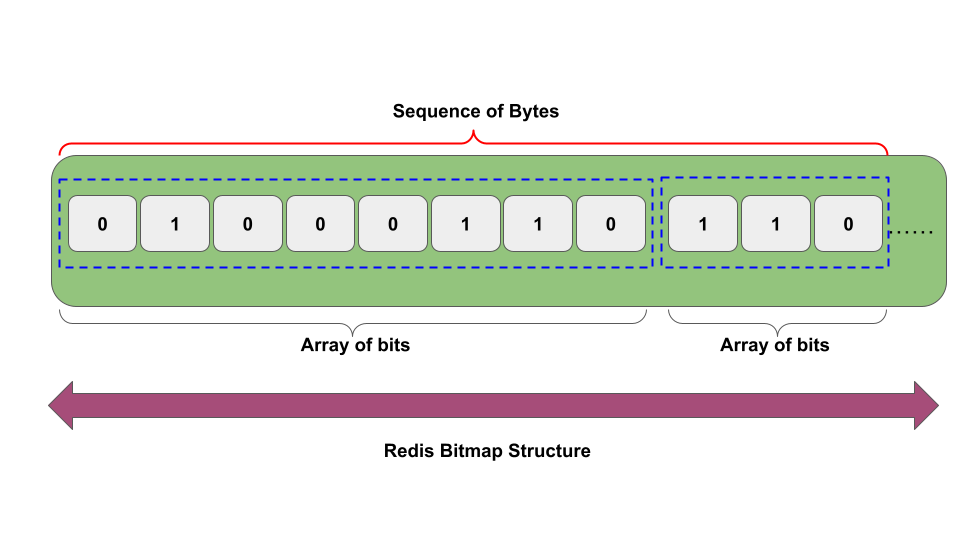
# 

# 

# 

# Redis Strings and Bitmaps

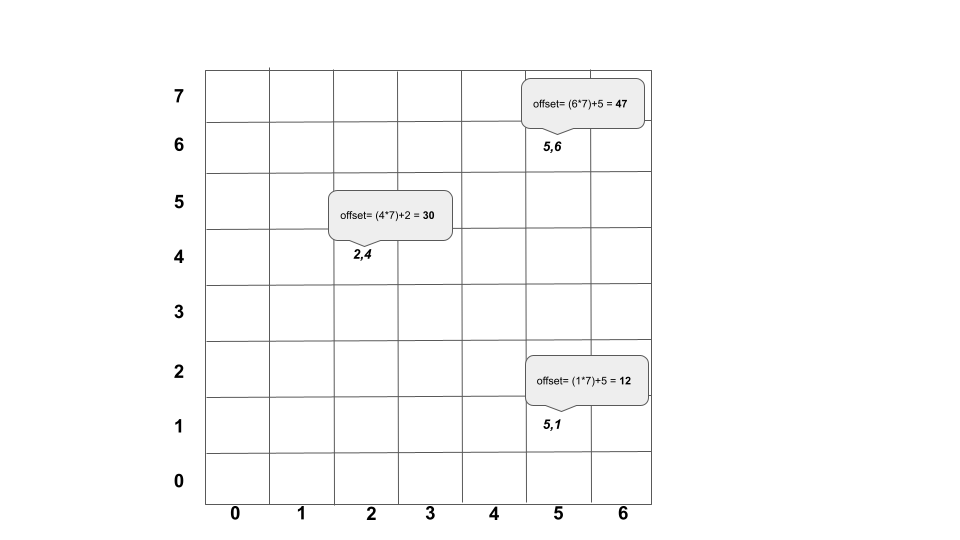
The string is the most fundamental Redis data type which can hold JSON objects, byte sequences, an array of binary values, and simple texts. The most influential fact about Redis strings is that they can be treated as a vector of bits. Hence, the Redis strings can be used to represent bitmaps. In short, a Redis bitmap can be seen as an array of bits stored in a Redis string.



Each bit in the bitmap is located using an offset value. The offset value of a given bit is calculated based on the x,y coordinates. The following mathematical formula can be applied to calculate the offset of a given bit.

| Offset = Y coordinate \* max\_width\_of\_map + X coordinate |
| --- |

Let’s consider the following bitmap and what the offset value looks like for different bits.



The values on each bit can be set using the offset value. Redis provides the SETBIT command to set a bit value to 0 or 1. In this Guide, we will be discussing the syntax and use cases of the SETBIT command.

# The SETBIT Command

The SETBIT command is used to manipulate bits stored at a given offset in a Redis string. Specifically, this command sets a given bit’s value to 1 or 0. Redis bitmap can hold up to 2^32-1 offsets which limits the maximum size of a bitmap to 512MB. Furthermore, the bitmap offsets start from the 0th index.

The following is the syntax of the SETBIT command.

| SETBIT key offset value |
| --- |

***key:*** The name of the Redis key which stores a string value.

***offset:*** The offset value of the bit.

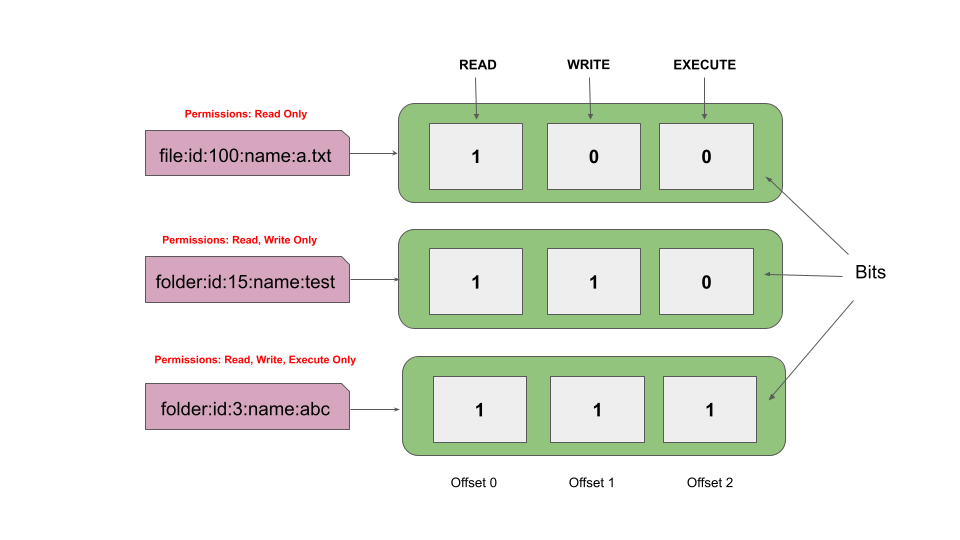
***value:*** The value of the bit. This can be 1 or 0.

If the key doesn’t exist, the SETBIT command will create a new string value and set the bit value at a specified offset. Furthermore, the SETBIT command operates on O(1) time complexity which is a considerably fast process. This might slightly vary when we are going to set the 2^31-1 bit in a bitmap when the specified key doesn’t exist or stores a very small string value. Because the memory for all the intermediate bits should be allocated. It would take some time.

The SETBIT command returns the original bit value stored at the offset in the string stored at a specified key.

## Use Case - File System Permissions Using Redis Bitmap

Let’s assume that a UNIX-like operating system maintains file permissions using bitmaps. The READ, WRITE and EXECUTE permissions for a given file/folder are stored in a bitmap as follows.



Let’s set the Read-only permissions to the file named “a.txt”. To make it clear, Read permission is denoted by the bit located at the 0th offset. So, we will be setting the 0th offset to 1 by calling the SETBIT command as follows.

| setbit file:id:100:name:a.txt 0 1 |
| --- |

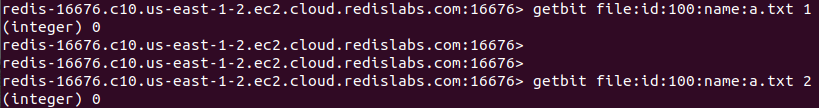


Let’s verify whether the 0th offset has been properly set using the GETBIT command as shown in the following.

| getbit file:id:100:name:a.txt 0 |
| --- |



We do not need to set the other two offsets explicitly since the default bit value of a bit is 0. Hence, the 1st and 2nd offset bits should be implicitly set to 0 as shown in the following.

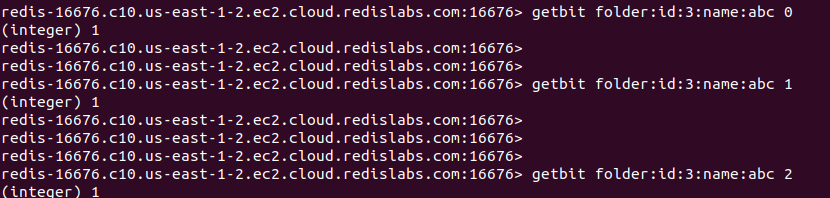


Similarly, we can set all three read, write, and execute permissions on the folder abc as follows. Multiple bits can be set by calling the SETBIT command multiple times.

| setbit folder:id:3:name:abc 0 1 setbit folder:id:3:name:abc 1 1 setbit folder:id:3:name:abc 2 1 |
| --- |

Let’s check the values of 3 bits using the GETBIT command.

| getbit folder:id:3:name:abc 0 getbit folder:id:3:name:abc 1 getbit folder:id:3:name:abc 2 |
| --- |



Furthermore, we can use the Redis GET command to retrieve the string representation of a given bitmap. As you can see in the following output, the hex value has been returned.



The equal decimal value would be 224.

Overall, the SETBIT command can be effectively used to manipulate the bit of bitmap.

# Conclusion

In summary, the SETBIT command is used to clear or set the bit value at a given offset in a Redis string stored at a given key. As mentioned, a Redis string can be treated as an array of bits that can represent a bitmap data structure. Also, the SETBIT command operates on bitmaps in O(1) time complexity. As you have seen in the use cases, multiple bits can be set by calling the SETBIT command multiple times. With the help of the Redis GET command, the string value of the bitmap can be retrieved.