



## FILE OPERATIONS.

මෙතෙක් සි වැඩසටහන භාවිතා කරන මෙහෙයුම් සිදු කරනු ලබන්නේ කිසිම තැනක ගබඩා නොවන ක්ෂණික / පර්යන්තයක ය. නමුත් මෘදුකාංග කර්මාන්තය තුළ, බොහෝ වැඩසටහන් ලියා ඇත්තේ වැඩසටහනෙන් ලබාගත් තොරතුරු ගබඩා කිරීම සඳහා ය. එවැනි එක් ක්‍රමයක් වන්නේ ලබාගත් තොරතුරු ගොනුවක ගබඩා කිරීමයි. ගොනුවක සිදු කළ හැකි විවිධ මෙහෙයුම්.

- A. Creation of a new file (fopen with attributes as "a" or "a+" or "w" or "w++")-නව ගොනුවක් සෑදීම ("a" හෝ "a +" හෝ "w" හෝ "w ++" ලෙස ගුණාංග සහිතව)
- B. Opening an existing file (fopen)- පවතින ගොනුවක් විවෘත කිරීම (fopen)
- C. Reading from file (fscanf or fgets)- ගොනුවෙන් කියවීම (fscanf හෝ fgets)
- D. Writing to a file (fprintf or fputs)- ගොනුවකට ලිවීම (fprintf හෝ fputs)
- E. Moving to a specific location in a file (fseek, rewind)- ගොනුවක නිශ්චිත ස්ථානයකට ගමන් කිරීම (fseek, rewind)
- F. Closing a file (fclose)- ගොනුවක් වැසීම (fclose)

Functions in File Operations:

File operation	Declaration & Description
<b>fopen() - To open a file</b>	<p>Declaration: FILE *fopen (const char *filename, const char *mode)</p> <p>fopen() function is used to open a file to perform operations such as reading, writing etc. In a C program, we declare a file pointer and use fopen() as below. fopen() function creates a new file if the mentioned file name does not exist.</p> <pre>FILE *fp; fp=fopen ("filename", "mode");</pre> <p>Where, fp - file pointer to the data type "FILE". filename - the actual file name with full path of the file. mode - refers to the operation that will be performed on the file. Example: r, w, a, r+, w+ and a+. Please refer below the description for these mode of operations.</p>
<b>fclose() - To close a file</b>	<p>Declaration: int fclose(FILE *fp);</p> <p>fclose() function closes the file that is being pointed by file pointer fp. In a C program, we close a file as below.</p> <pre>fclose (fp);</pre>
<b>fgets() - To read a file</b>	<p>Declaration: char *fgets(char *string, int n, FILE *fp)</p> <p>fgets function is used to read a file line by line. In a C program, we use fgets function as below.</p> <pre>fgets (buffer, size, fp);</pre> <p>where, buffer - buffer to put the data in. size - size of the buffer fp - file pointer</p>
<b>fprintf() - To write into a file</b>	<p>Declaration:</p> <pre>int fprintf(FILE *fp, const char *format, ...);</pre> <p>fprintf() function writes string into a file pointed by fp. In a C program, we write string into a file as below. fprintf (fp, "some data"); or fprintf (fp, "text %d", variable_name);</p>

## Opening or creating file

ගොනුවක් විවෘත කිරීම සඳහා, අවශ්‍ය ප්‍රවේශ ක්‍රම සමඟ **fopen** ශ්‍රිතය භාවිතා කරයි. බහුලව භාවිතා වන ගොනු ප්‍රවේශ ක්‍රම කිහිපයක් පහත දැක්වේ.

- "r" - Searches file. If the file is opened successfully fopen() loads it into memory and sets up a pointer which points to the first character in it. If the file cannot be opened fopen() returns NULL.
- "w" - Searches file. If the file exists, its contents are overwritten. If the file doesn't exist, a new file is created. Returns NULL, if unable to open file.

- c. "a" – Searches file. If the file is opened successfully fopen() loads it into memory and sets up a pointer that points to the last character in it. If the file doesn't exist, a new file is created. Returns NULL, if unable to open file.
- d. "r+" – Searches file. If is opened successfully fopen() loads it into memory and sets up a pointer which points to the first character in it. Returns NULL, if unable to open the file.
- e. "w+" – Searches file. If the file exists, its contents are overwritten. If the file doesn't exist a new file is created. Returns NULL, if unable to open file.
- f. "a+" – Searches file. If the file is opened successfully fopen() loads it into memory and sets up a pointer which points to the last character in it. If the file doesn't exist, a new file is created. Returns NULL, if unable to open file

r	ගොනුව සොයයි. ගොනුව සාර්ථකව විවෘත කළහොත් <b>fopen ()</b> එය මතකය වෙත පටවා එහි පළමු අක්ෂරයට යොමු වන දර්ශකයක් සකසයි. ගොනුව විවෘත කළ නොහැකි නම් <b>fopen ()</b> NULL ලබා දෙයි
w	ගොනුව සොයයි. ගොනුව තිබේ නම්, එහි අන්තර්ගතය නැවත ලියනු ලැබේ. ගොනුව නොපවතී නම්, නව ගොනුවක් සාදනු ලැබේ. ගොනුව විවෘත කිරීමට නොහැකි නම් NULL ලබා දෙයි
a	ගොනුව සොයයි. ගොනුව සාර්ථකව විවෘත කර ඇත්නම් <b>fopen ()</b> එය මතකයට පටවා එහි ඇති අවසාන අක්ෂරයට යොමු වන දර්ශකයක් සකසයි. ගොනුව නොපවතී නම්, නව ගොනුවක් සාදනු ලැබේ. ගොනුව විවෘත කිරීමට නොහැකි නම් NULL ලබා දෙයි
r+	ගොනුව සොයයි. සාර්ථකව විවෘත කළහොත් <b>fopen ()</b> එය මතකය තුළට පටවා එහි පළමු අක්ෂරයට යොමු වන දර්ශකයක් සකසයි. ගොනුව විවෘත කිරීමට නොහැකි නම් NULL ලබා දෙයි
w+	ගොනුව සොයයි. ගොනුව තිබේ නම්, එහි අන්තර්ගතය නැවත ලියනු ලැබේ. ගොනුව නොපවතී නම් නව ගොනුවක් සාදනු ලැබේ. ගොනුව විවෘත කිරීමට නොහැකි නම් NULL ලබා දෙයි.
a+	ගොනුව සොයයි. ගොනුව සාර්ථකව විවෘත කර ඇත්නම් <b>fopen ()</b> එය මතකයට පටවා එහි ඇති අවසාන අක්ෂරය පෙන්නා දෙන දර්ශකයක් සකසයි. ගොනුව නොපවතී නම්, නව

	ගොනුවක් සාදනු ලැබේ. ගොනුව විවෘත කිරීමට නොහැකි නම් NULL ලබා දෙයි
--	---

ඉහත දක්වා ඇති පරිදි, ඔබට ද්විමය ගොනුවක මෙහෙයුම් සිදු කිරීමට අවශ්‍ය නම්, ඔබ අවසානයේ දී 'b' එකතු කළ යුතුය. උදාහරණයක් ලෙස, "w" වෙනුවට "wb" භාවිතා කළ යුතු අතර "a +" වෙනුවට "a + b" භාවිතා කළ යුතුය. ගොනුවේ මෙහෙයුම් සිදු කිරීම සඳහා, ගොනු දර්ශකය නමින් විශේෂ දර්ශකයක් භාවිතා කරනු ලැබේ

(As given above, if you want to perform operations on a binary file, then you have to append 'b' at the last. For example, instead of "w", you have to use "wb", instead of "a+" you have to use "a+b". For performing the operations on the file, a special pointer called File pointer is used which is declared as:)

```
FILE *filePointer;

So, the file can be opened as

filePointer = fopen("file.txt", "w")
```

## Reading from a file

Fscanf හෝ fgets භාවිතයෙන් ගොනු කියවීමේ මෙහෙයුම් සිදු කළ හැකිය. මෙම කාර්යයන් දෙකම ස්කෑන් හා සමාන ක්‍රියාකාරිත්වයන් සිදු කළ නමුත් අතිරේක පරාමිතියක් සමඟ ගොනු දර්ශකය. එබැවින්, ඔබට ගොනු රේඛාව රේඛාවෙන් හෝ අක්ෂර අනුව කියවීමට අවශ්‍ය නම් එය ඔබ මත රඳා පවතී.

And the code snippet for reading a file is as:

```
FILE * filePointer;

filePointer = fopen("fileName.txt", "r");

fscanf(filePointer, "%s %s %s %d", str1, str2, str3, &year);
```

## Writing a file

කියවීමේ මෙහෙයුම් වලට සමානකම් ඇති fprintf සහ fputs මගින් ගොනු ලිවීමේ මෙහෙයුම් පරිපූර්ණ කළ හැකිය.

```
FILE *filePointer ;

filePointer = fopen("fileName.txt", "w");

fprintf(filePointer, "%s %s %s %d", "We", "are", "in", 2012);
```

## Closing a file

සෑම සාර්ථක file මෙහෙයුමකින් පසු, ඔබ සැමවිටම ගොනුවක් වසා දැමිය යුතුය. ගොනුවක් වැසීම සඳහා, ඔබට fclose ශ්‍රිතය භාවිතා කළ යුතුය.

```
FILE *filePointer ;  
  
filePointer= fopen("fileName.txt", "w");  
  
----- Some file Operations -----  
  
fclose(filePointer)
```