FILE HANDLING FUNCTIONS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Functions | Description | Return | Syntax | Example |
| fopen | Opens a file | Opens the file whose name is specified in the parameter filename and associates it with a stream that can be identified in future operations by the FILE pointer returned. | FILE \* fopen ( const char \* filename, const char \* mode ); | FILE \*fptr;  fptr = fopen(“filename.txt”, “mode”); |
| fclose | Closes a file | If the stream is successfully closed, a zero value is returned.  On failure, EOF is returned. | int fclose ( FILE \* stream ); | fclose(fptr); |
| fgetc | Reads character from file | On success, the character read is returned (promoted to an int value).  The return type is int to accommodate for the special value EOF, which indicates failure: | int fgetc ( FILE \* stream ); | int ch;  while((ch = fgetc(fptr)) != EOF); |
| fgets | Get string from stream | On success, the function returns str. | char \* fgets ( char \* str, int num, FILE \* stream ); | Char buffer[10];  fgets(buffer, 256, fptr); |
| fputc | Write character to stream | On success, the character written is returned. | int fputc ( int character, FILE \* stream ); | int ch;  for( ch = 33 ; ch <= 100; ch++ ) {  putc(ch, fp);  } |
| fputs | Write string to stream | On success, a non-negative value is returned.  On error, the function returns EOF | int fputs ( const char \* str, FILE \* stream ); | fputs("This is c programming.", fp); |
| ftell | Get current position in stream | On success, the current value of the position indicator is returned.  On failure, -1L is returned, and errno is set to a system-specific positive value. | long int ftell ( FILE \* stream ); | ftell(fptr); |
| fflush | Flush stream |  | int fflush ( FILE \* stream ); |  |
| fseek | Reposition stream position indicator | If successful, the function returns zero.  Otherwise, it returns non-zero value. | int fseek ( FILE \* stream, long int offset, int origin ); | fseek(fptr, 10, SEEK\_SET/ SEEK\_END, SEEK\_CUR); |
| rewind | Set position of stream to the beginning |  | void rewind ( FILE \* stream ); | rewind(fptr); |
| feof | Check end-of-file indicator | A non-zero value is returned in the case that the end-of-file indicator associated with the stream is set.  Otherwise, zero is returned. | int feof ( FILE \* stream ); | foef(fptr); |
| ferror | Check error indicator | A non-zero value is returned in the case that the error indicator associated with the stream is set.  Otherwise, zero is returned. | int ferror ( FILE \* stream ); | c = fgetc(fp);  if( ferror(fp) ) {  printf("Error in reading from file : file.txt\n");  } |
| remove | Remove file | If the file is successfully deleted, a zero value is returned.  On failure, a nonzero value is returned. | int remove ( const char \* filename ); | ret = remove(filename);  if(ret == 0) {  printf("File deleted successfully");  } else {  printf("Error: unable to delete the file");  } |
| rename | Rename file | If the file is successfully renamed, a zero value is returned.  On failure, a nonzero value is returned. | int rename ( const char \* oldname, const char \* newname ); | char old\_name[] = "old.txt";  char new\_name[] = “new.txt”;  rename(old\_name, new\_name); |
| getline  <https://linux.die.net/man/3/getline> | return the number of characters read | Return the no of length of the line from the file stream | ssize\_t getline(char \*\*lineptr, size\_t \*n, FILE \*stream);  ssize\_t getdelim(char \*\*lineptr, size\_t \*n, int delim, FILE \*stream); | char \*line;  size\_t n;  getline(&line, &n, fptr); |