My Project

Generated by Doxygen 1.15.0

Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 2

File Documentation

2.1 demo.c File Reference

Demonstration of the UrduC library functions and macros.

```
#include <urdu.h>
```

Functions

- int **main** (void)

2.1.1 Detailed Description

Demonstration of the UrduC library functions and macros.

This program demonstrates all the primary features of the UrduC library:

- Input/output wrappers
- String manipulation
- Character handling
- File operations
- Control structure macros and loops

2.2 urdu.h File Reference

UrduC Library — C language made easier and friendlier with Urdu-style names.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

Macros

- #define **agr** if

  Equivalent of `if`*.*
- #define **warna** else

  Equivalent of `else`*.*
- #define **warna_agr** else if

  Equivalent of `else if`*.*
- #define **badal** switch

  Equivalent of `switch`*.*
- #define **surat** case

  Equivalent of `case`*.*
- #define **warna_sab** default

  Equivalent of `default`*.*
- #define jab_tak(condition)

  Equivalent of `while`*.*
- #define **karo** do

  Equivalent of `do`*.*
- #define jab_(condition)

  Equivalent of `while` *(used with* `do`*).*
- #define **agay_chalo** continue

  Equivalent of `continue`*.*
- #define **ruk_jao** break

  Equivalent of `break`*.*
- #define file_kholo(filename, mode)

  Opens a file.
- #define file_band(stream)

  Closes a file.
- #define file_likho(stream, ...)

  Writes formatted data to a file.
- #define file_lo(stream, ...)

  Reads formatted data from a file.
- #define line_lo(str, size, stream)

  Reads a line from a file.
- #define line_likho(str, stream)

  Writes a line to a file.
- #define bara_karo(ch)

  Converts a character to uppercase.
- #define chhota_karo(ch)

  Converts a character to lowercase.
- #define kya_adad(ch)

  Checks if a character is a digit.
- #define kya_harf(ch)

  Checks if a character is an alphabetic letter.
- #define kya_harfya_adad(ch)

  Checks if a character is alphanumeric.

Functions

- int **parhle_int** (const char ∗prompt)
  Reads an integer value from the user with a prompt.
- void parhle (const char ∗prompt, char ∗buf, int size)
  Reads a line of text safely into a buffer.
- float **parhle_float** (const char ∗prompt)
  Reads a float value from user input.
- double **parhle_double** (const char ∗prompt)
  Reads a double value from user input.
- void **likhde** (const char ∗text)
  Prints text to the console.
- void **likhde_int** (int n)
  Prints an integer.
- void **likhde_float** (float n)
  Prints a float with two decimal precision.
- void **likhde_double** (double n)
  Prints a double with two decimal precision.
- void **likhde_line** (const char ∗text)
  Prints text followed by a newline.
- void **likhde_int_line** (int n)
  Prints an integer followed by a newline.
- void **likhde_float_line** (float n)
  Prints a float followed by a newline.
- void **likhde_harf** (char ch)
  Prints a single character.
- void **likhde_harf_line** (char ch)
  Prints a single character followed by a newline.
- char ∗ **jodo** (char ∗dest, const char ∗src)
  Concatenates two strings (equivalent to strcat).
- char ∗ **nakal** (char ∗dest, const char ∗src)
  Copies one string to another (equivalent to strcpy).
- int **muqabla** (const char ∗s1, const char ∗s2)
  Compares two strings (equivalent to strcmp).
- int **lambai** (const char ∗str)
  Returns the length of a string (equivalent to strlen).

2.2.1 Detailed Description

UrduC Library — C language made easier and friendlier with Urdu-style names.

This header provides a collection of macros and functions designed to make learning and using C programming more intuitive by mapping common functions and keywords into Urdu transliteration equivalents.

The library includes:

- Input/Output functions
- String manipulation functions
- Character utilities
- Control structure macros (if, else, loops)
- File handling wrappers

Author

Version

1.0

Date

2025

2.2.2 Macro Definition Documentation

2.2.2.1 bara_karo

```
#define bara_karo(
              ch)
```

Value:

```
toupper(ch)
```

Converts a character to uppercase.

2.2.2.2 chhota_karo

```
#define chhota_karo(
              ch)
```

Value:

```
tolower(ch)
```

Converts a character to lowercase.

2.2.2.3 file_band

```
#define file_band(
              stream)
```

Value:

```
fclose(stream)
```

Closes a file.

2.2.2.4 file_kholo

```
#define file_kholo(
               filename,
               mode)
```

Value:

```
fopen(filename, mode)
```

Opens a file.

2.2.2.5 file_likho

```
#define file_likho(
               stream,
               ...)
```

Value:

```
fprintf(stream, __VA_ARGS__)
```

Writes formatted data to a file.

2.2.2.6 file_lo

```
#define file_lo(
               stream,
               ...)
```

Value:

```
fscanf(stream, __VA_ARGS__)
```

Reads formatted data from a file.

2.2.2.7 jab_

```
#define jab_(
               condition)
```

Value:

```
while (condition)
```

Equivalent of `while` (used with `do`).

2.2.2.8 jab_tak

```
#define jab_tak(
               condition)
```

Value:

```
while (condition)
```

Equivalent of `while`.

2.2.2.9 kya_adad

```
#define kya_adad(
              ch)
```

Value:

```
isdigit(ch)
```

Checks if a character is a digit.

2.2.2.10 kya_harf

```
#define kya_harf(
              ch)
```

Value:

```
isalpha(ch)
```

Checks if a character is an alphabetic letter.

2.2.2.11 kya_harfya_adad

```
#define kya_harfya_adad(
              ch)
```

Value:

```
isalnum(ch)
```

Checks if a character is alphanumeric.

2.2.2.12 line_likho

```
#define line_likho(
              str,
              stream)
```

Value:

```
fputs(str, stream)
```

Writes a line to a file.

2.2.2.13 line_lo

```
#define line_lo(
              str,
              size,
              stream)
```

Value:

```
fgets(str, size, stream)
```

Reads a line from a file.

2.2.3 Function Documentation

2.2.3.1 parhle()

```
void parhle (
            const char * prompt,
            char * buf,
            int size)
```

Reads a line of text safely into a buffer.

Parameters

prompt	The message displayed to the user.
buf	The destination buffer.
size	The buffer size.

2.3 urdu.h

Go to the documentation of this file.

```
00001
00019
00020 #ifndef URDU_H
00021 #define URDU_H
00022
00023 #include <stdio.h>
00024 #include <stdlib.h>
00025 #include <string.h>
00026 #include <ctype.h>
00027
00029 int parhle_int(const char *prompt);
00030
00037 void parhle(const char *prompt, char *buf, int size);
00038
00040 float parhle_float(const char *prompt);
00041
00043 double parhle_double(const char *prompt);
00044
00046 void likhde(const char *text);
00047
00049 void likhde_int(int n);
00050
00052 void likhde_float(float n);
00053
00055 void likhde_double(double n);
00056
00058 void likhde_line(const char *text);
00059
00061 void likhde_int_line(int n);
00062
00064 void likhde_float_line(float n);
00065
00067 void likhde_harf(char ch);
00068
00070 void likhde_harf_line(char ch);
00071
00073 char* jodo(char* dest, const char* src);
00074
00076 char* nakal(char* dest, const char* src);
00077
00079 int muqabla(const char* s1, const char* s2);
00080
00082 int lambai(const char* str);
00083
00084 /* ================================================================
00085    CONTROL STRUCTURE MACROS
00086    ================================================================ */
00087
00089 #define agr if
00090
00092 #define warna else
00093
00095 #define warna_agr else if
00096
00098 #define badal switch
00099
00101 #define surat case
00102
00104 #define warna_sab default
00105
00107 #define jab_tak(condition) while (condition)
00108
00110 #define karo do
00111
00113 #define jab_(condition) while (condition)
00114
00116 #define agay_chalo continue
00117
00119 #define ruk_jao break
00120
00121 /* ================================================================
00122    FILE HANDLING MACROS
```

```
00123    ========================================================================== */
00124
00126 #define file_kholo(filename, mode) fopen(filename, mode)
00127
00129 #define file_band(stream) fclose(stream)
00130
00132 #define file_likho(stream, ...) fprintf(stream, __VA_ARGS__)
00133
00135 #define file_lo(stream, ...) fscanf(stream, __VA_ARGS__)
00136
00138 #define line_lo(str, size, stream) fgets(str, size, stream)
00139
00141 #define line_likho(str, stream) fputs(str, stream)
00142
00143 /* ========================================================================
00144    CHARACTER HANDLING MACROS
00145    ========================================================================== */
00146
00148 #define bara_karo(ch) toupper(ch)
00149
00151 #define chhota_karo(ch) tolower(ch)
00152
00154 #define kya_adad(ch) isdigit(ch)
00155
00157 #define kya_harf(ch) isalpha(ch)
00158
00160 #define kya_harfya_adad(ch) isalnum(ch)
00161
00162 #endif /* URDU_H */
```

Index