

Practice-C

1.0.0

다음에 의해 생성됨 : Doxygen 1.8.13

Contents

1	데이터 구조 색인	1
1.1	데이터 구조	1
2	파일 색인	3
2.1	파일 목록	3
3	데이터 구조 문서화	5
3.1	_struct 구조체 참조	5
3.1.1	상세한 설명	5
3.1.2	필드 문서화	5
3.1.2.1	i	6
3.2	data_st 구조체 참조	6
3.2.1	상세한 설명	6
3.2.2	필드 문서화	6
3.2.2.1	age	7
3.2.2.2	ch	7
3.2.2.3	name	7
3.3	inventory 구조체 참조	7
3.3.1	상세한 설명	8
3.3.2	필드 문서화	8
3.3.2.1	leadtime	8
3.3.2.2	name	8
3.3.2.3	number	8
3.3.2.4	volume	8

4 파일 문서화	9
4.1 src/function/base.c 파일 참조	9
4.1.1 상세한 설명	10
4.1.2 매크로 문서화	10
4.1.2.1 BUFF_SIZE	10
4.1.3 타입정의 문서화	10
4.1.3.1 STRUCT	10
4.1.4 함수 문서화	10
4.1.4.1 main()	11
4.2 src/libstd/base.c 파일 참조	11
4.2.1 상세한 설명	12
4.2.2 매크로 문서화	12
4.2.2.1 BUFF_SIZE	12
4.2.3 타입정의 문서화	12
4.2.3.1 STRUCT	12
4.2.4 함수 문서화	12
4.2.4.1 main()	12
4.3 src/function/Declaring_Attributes_of_Functions.c 파일 참조	13
4.3.1 함수 문서화	13
4.3.1.1 __func_noreturn()	13
4.3.1.2 __func_visibility()	14
4.3.1.3 alias()	14
4.3.1.4 debug()	15
4.3.1.5 f_nonnull()	15
4.3.1.6 f_noreturn()	15
4.3.1.7 f_visibility()	16
4.3.1.8 f_warn_unused_result()	16
4.3.1.9 f_weakref()	16
4.4 src/function/Declaring_Attributes_of_Functions.h 파일 참조	16
4.4.1 함수 문서화	17

4.4.1.1	<code>__func_nonnull()</code>	17
4.4.1.2	<code>__func_noreturn()</code>	17
4.4.1.3	<code>__func_visibility()</code>	18
4.4.1.4	<code>__func_warn_unused_result()</code>	18
4.4.1.5	<code>__func_weakref()</code>	18
4.4.1.6	<code>debug()</code>	18
4.4.1.7	<code>error()</code>	19
4.4.1.8	<code>f_nonnull()</code>	19
4.4.1.9	<code>f_noreturn()</code>	19
4.4.1.10	<code>f_visibility()</code>	19
4.4.1.11	<code>f_warn_unused_result()</code>	19
4.4.1.12	<code>f_weakref()</code>	20
4.5	<code>src/function/Declaring_Attributes_of_Functions2.c</code> 파일 참조	20
4.5.1	함수 문서화	20
4.5.1.1	<code>f_external()</code>	20
4.6	<code>src/function/Declaring_Attributes_of_Functions_main.c</code> 파일 참조	20
4.6.1	함수 문서화	21
4.6.1.1	<code>stf_weakref()</code>	21
4.6.2	변수 문서화	21
4.6.2.1	<code>p_func</code>	21
4.7	<code>src/function/main.c</code> 파일 참조	21
4.7.1	상세한 설명	22
4.7.2	매크로 문서화	22
4.7.2.1	<code>ARRAY_SIZE</code>	22
4.7.2.2	<code>BUFF_SIZE</code>	23
4.7.3	타입정의 문서화	23
4.7.3.1	<code>INVENTORY</code>	23
4.7.4	함수 문서화	23
4.7.4.1	<code>displInventory()</code>	23
4.7.4.2	<code>main()</code>	24

4.7.4.3	operationError()	24
4.7.4.4	sleep()	25
4.8	src/pointer/main.c 파일 참조	25
4.8.1	매크로 문서화	26
4.8.1.1	COL_MAX	26
4.8.1.2	ROW_MAX	26
4.8.2	함수 문서화	26
4.8.2.1	main()	27
4.8.2.2	vaPrintf()	27
4.8.3	변수 문서화	27
4.8.3.1	a	27
4.9	src/keyword/structure.c 파일 참조	28
4.9.1	상세한 설명	28
4.9.2	매크로 문서화	29
4.9.2.1	BUFF_SIZE	29
4.9.3	타입정의 문서화	29
4.9.3.1	STRUCT	29
4.9.4	함수 문서화	29
4.9.4.1	main()	29
4.10	src/libstd/abort.c 파일 참조	30
4.10.1	상세한 설명	30
4.10.2	매크로 문서화	31
4.10.2.1	BUFF_SIZE	31
4.10.3	타입정의 문서화	31
4.10.3.1	STRUCT	31
4.10.4	함수 문서화	31
4.10.4.1	myAbort()	31
4.11	src/libstd/clock.c 파일 참조	31
4.11.1	상세한 설명	32
4.11.2	매크로 문서화	33

4.11.2.1	BUFF_SIZE	33
4.11.3	타입정의 문서화	33
4.11.3.1	DATA	33
4.11.4	함수 문서화	33
4.11.4.1	myMemcpy()	33
4.12	src/libstd/common.h 파일 참조	34
4.13	src/pointer/ptr.c 파일 참조	34
4.13.1	매크로 문서화	35
4.13.1.1	STACK_SIZE	35
4.13.2	타입정의 문서화	35
4.13.2.1	stack_t	35
4.13.3	함수 문서화	35
4.13.3.1	pop()	36
4.13.3.2	push()	36
4.13.4	변수 문서화	36
4.13.4.1	stack	36
4.13.4.2	top	36
4.14	src/pointer/ptr.h 파일 참조	36
색인		37

Chapter 1

데이터 구조 색인

1.1 데이터 구조

다음은 데이터 구조들입니다. (간략한 설명만을 보여줍니다) :

_struct	5
data_st	6
inventory	7

Chapter 2

파일 색인

2.1 파일 목록

다음은 모든 파일에 대한 목록입니다. (간략한 설명만을 보여줍니다) :

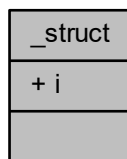
src/function/ base.c	9
src/function/ Declaring_Attributes_of_Functions.c	13
src/function/ Declaring_Attributes_of_Functions.h	16
src/function/ Declaring_Attributes_of_Functions2.c	20
src/function/ Declaring_Attributes_of_Functions_main.c	20
src/function/ main.c	21
src/keyword/ structure.c	28
src/libstd/ abort.c	30
src/libstd/ base.c	11
src/libstd/ clock.c	31
src/libstd/ common.h	34
src/pointer/ main.c	25
src/pointer/ ptr.c	34
src/pointer/ ptr.h	36

Chapter 3

데이터 구조 문서화

3.1 `_struct` 구조체 참조

`_struct`에 대한 협력 다이어그램:



데이터 필드

- `int i`

3.1.1 상세한 설명

`base.c` 파일의 30 번째 라인에서 정의되었습니다.

3.1.2 필드 문서화

3.1.2.1 i

```
int i
```

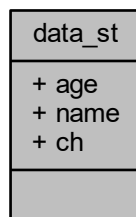
base.c 파일의 31 번째 라인에서 정의되었습니다.

이 구조체에 대한 문서화 페이지는 다음의 파일들로부터 생성되었습니다.:

- src/function/[base.c](#)
- src/keyword/[structure.c](#)
- src/libstd/[abort.c](#)

3.2 data_st 구조체 참조

data_st에 대한 협력 다이어그램:



데이터 필드

- int32_t [age](#)
- int8_t [name](#) [25]
- int8_t [ch](#)

3.2.1 상세한 설명

clock.c 파일의 30 번째 라인에서 정의되었습니다.

3.2.2 필드 문서화

3.2.2.1 age

```
int32_t age
```

clock.c 파일의 31 번째 라인에서 정의되었습니다.

3.2.2.2 ch

```
int8_t ch
```

clock.c 파일의 33 번째 라인에서 정의되었습니다.

3.2.2.3 name

```
int8_t name[25]
```

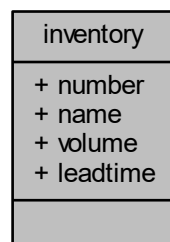
clock.c 파일의 32 번째 라인에서 정의되었습니다.

이 구조체에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- src/libstd/[clock.c](#)

3.3 inventory 구조체 참조

inventory에 대한 협력 다이어그램:



데이터 필드

- char * [number](#)
상품 번호
- char * [name](#)
상품명
- int [volume](#)
재고 수량
- int [leadtime](#)
매입 일수

3.3.1 상세한 설명

상품 재고 관리용 inventory 구조체 정의
main.c 파일의 31 번째 라인에서 정의되었습니다.

3.3.2 필드 문서화

3.3.2.1 leadtime

```
int leadtime
```

매입 일수

main.c 파일의 35 번째 라인에서 정의되었습니다.

3.3.2.2 name

```
char* name
```

상품명

main.c 파일의 33 번째 라인에서 정의되었습니다.

3.3.2.3 number

```
char* number
```

상품 번호

main.c 파일의 32 번째 라인에서 정의되었습니다.

3.3.2.4 volume

```
int volume
```

재고 수량

main.c 파일의 34 번째 라인에서 정의되었습니다.

이 구조체에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- [src/function/main.c](#)

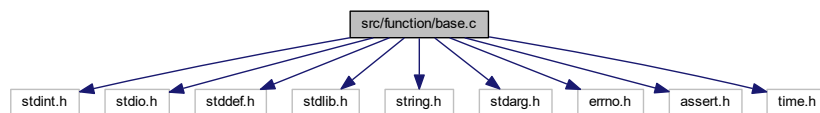
Chapter 4

파일 문서화

4.1 src/function/base.c 파일 참조

```
#include <stdint.h>
#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <errno.h>
#include <assert.h>
#include <time.h>
```

base.c에 대한 include 의존 그래프



데이터 구조

- `struct _struct`

매크로

- `#define BUFF_SIZE 256`
Buffer size for character data

타입정의

- `typedef struct _struct STRUCT`

함수

- int `main` (int argc, char const *argv[])

4.1.1 상세한 설명

작성자

날짜

참고

4.1.2 매크로 문서화

4.1.2.1 `BUFF_SIZE`

```
#define BUFF_SIZE 256
```

Buffer size for character data

base.c 파일의 22 번째 라인에서 정의되었습니다.

4.1.3 타입정의 문서화

4.1.3.1 `STRUCT`

```
typedef struct _struct STRUCT
```

4.1.4 함수 문서화

4.1.4.1 main()

```
int main (
    int argc,
    char const * argv[] )
```

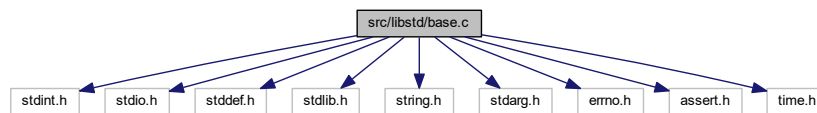
base.c 파일의 54 번째 라인에서 정의되었습니다.

```
54                                     {
55
56     return 0;
57 }
```

4.2 src/libstd/base.c 파일 참조

```
#include <stdint.h>
#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <errno.h>
#include <assert.h>
#include <time.h>
```

base.c에 대한 include 의존 그래프



데이터 구조

- struct [_struct](#)

매크로

- #define [BUFF_SIZE](#) 256
Buffer size for character data

타입정의

- typedef struct [_struct](#) [STRUCT](#)

함수

- int [main](#) (int argc, char const *argv[])

4.2.1 상세한 설명

작성자

날짜

참고

4.2.2 매크로 문서화

4.2.2.1 BUFF_SIZE

```
#define BUFF_SIZE 256
```

Buffer size for character data

base.c 파일의 22 번째 라인에서 정의되었습니다.

4.2.3 타입정의 문서화

4.2.3.1 STRUCT

```
typedef struct _struct STRUCT
```

4.2.4 함수 문서화

4.2.4.1 main()

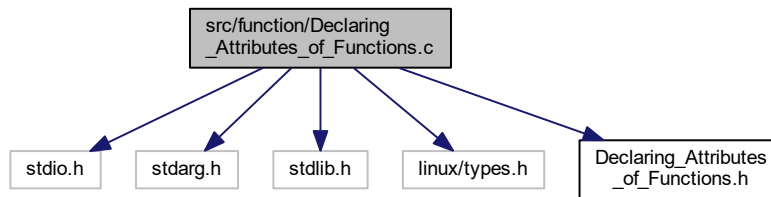
```
int main (
    int argc,
    char const * argv[] )
```

base.c 파일의 54 번째 라인에서 정의되었습니다.

```
54                                     {
55
56     return 0;
57 }
```

4.3 src/function/Declaring_Attributes_of_Functions.c 파일 참조

```
#include <stdio.h>
#include <stdarg.h>
#include <stdlib.h>
#include <linux/types.h>
#include "Declaring_Attributes_of_Functions.h"
Declaring_Attributes_of_Functions.c에 대한 include 의존 그래프
```



함수

- void `__func_noreturn` (void)
- void `f_noreturn` (void)
- void `f_nonnull` (void *dest, const void *src, size_t len) `__attribute__((weak`
- void `alias` ("__func_nonnull"))
- void `__func_visibility` (void)
- void `f_visibility` (void)
- int `f_warn_unused_result` (void)
- void `f_weakref` (void)
- void `debug` (__s32 dlevel, const char *fmt,...)

4.3.1 함수 문서화

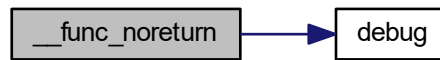
4.3.1.1 __func_noreturn()

```
void __func_noreturn (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 7 번째 라인에서 정의되었습니다.

```
8 {
9     debug(1, "%s is called...\n", __func__);
10    exit(1); /* or abort(); */
11 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



4.3.1.2 __func_visibility()

```
void __func_visibility (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 22 번째 라인에서 정의되었습니다.

```
23 {
24     debug(1, "%s is called..\n", __func__);
25 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



4.3.1.3 alias()

```
void alias (
    "__func_nonnull" )
```

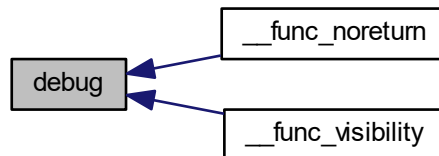
4.3.1.4 debug()

```
void debug (
    __s32 dlevel,
    const char * fmt,
    ... )
```

Declaring_Attributes_of_Functions.c 파일의 53 번째 라인에서 정의되었습니다.

```
54 {
55     va_list ap;
56     va_start (ap, fmt);
57
58     switch (dlevel) {
59     case 0:
60         break;
61     case 1:
62         vfprintf(stdout, fmt, ap);
63         va_end(ap);
64         break;
65     default:
66         vfprintf(stdout, fmt, ap);
67         va_end(ap);
68         break;
69     };
70 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



4.3.1.5 f_nonnull()

```
void f_nonnull (
    void * dest,
    const void * src,
    size_t len )
```

4.3.1.6 f_noreturn()

```
void f_noreturn (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 13 번째 라인에서 정의되었습니다.

```
16 {
17     debug(1, "%s is called...\n", __func__);
18 }
```

4.3.1.7 f_visibility()

```
void f_visibility (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 27 번째 라인에서 정의되었습니다.

```
31 {
32     debug(1, "%s is called..\n", __func__);
33     return 0;
34 }
```

4.3.1.8 f_warn_unused_result()

```
int f_warn_unused_result (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 36 번째 라인에서 정의되었습니다.

```
39 {
40     debug(1, "%s is called..\n", __func__);
41 }
```

4.3.1.9 f_weakref()

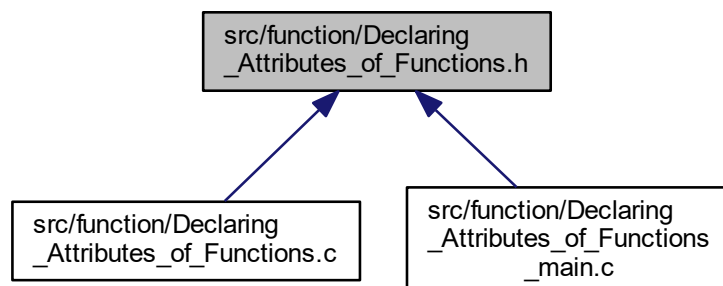
```
void f_weakref (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 42 번째 라인에서 정의되었습니다.

```
45 {
46     va_list ap;
47     va_start (ap, fmt);
48     vfprintf(stderr, fmt, ap);
49     va_end(ap);
50     va_end(ap);
51 }
```

4.4 src/function/Declaring_Attributes_of_Functions.h 파일 참조

이 그래프는 이 파일을 직/간접적으로 include 하는 파일들을 보여줍니다.:



함수

- void `f_noreturn` (void) __attribute__((noreturn))
- void `f_nonnull` (void *dest, const void *src, size_t len) __attribute__((nonnull))
- void `f_visibility` (void) __attribute__((visibility("hidden")))
- int `f_warn_unused_result` (void) __attribute__((warn_unused_result))
- void `f_weakref` (void)
- void `__func_noreturn` (void) __attribute__((noreturn))
- void `__func_nonnull` (void *dest, const void *src, size_t len) __attribute__((nonnull))
- void `__func_visibility` (void) __attribute__((visibility("hidden")))
- int `__func_warn_unused_result` (void) __attribute__((warn_unused_result))
- void `__func_weakref` (void)
- void `error` (const char *format,...) __attribute__((format(printf
- void void `debug` (int dlevel, const char *format,...) __attribute__((format(printf

4.4.1 함수 문서화

4.4.1.1 __func_nonnull()

```
void __func_nonnull (
    void * dest,
    const void * src,
    size_t len )
```

4.4.1.2 __func_noreturn()

```
void __func_noreturn (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 7 번째 라인에서 정의되었습니다.

```
8 {
9     debug(1, "%s is called.\n", __func__);
10     exit(1); /* or abort(); */
11 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



4.4.1.3 __func_visibility()

```
void __func_visibility (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 22 번째 라인에서 정의되었습니다.

```
23 {
24     debug(1, "%s is called..\n", __func__);
25 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



4.4.1.4 __func_warn_unused_result()

```
int __func_warn_unused_result (
    void )
```

4.4.1.5 __func_weakref()

```
void __func_weakref (
    void )
```

4.4.1.6 debug()

```
void void debug (
    int dlevel,
    const char * format,
    ... )
```

4.4.1.7 error()

```
void error (
    const char * format,
    ... )
```

4.4.1.8 f_nonnull()

```
void f_nonnull (
    void * dest,
    const void * src,
    size_t len )
```

4.4.1.9 f_noreturn()

```
void f_noreturn (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 13 번째 라인에서 정의되었습니다.

```
16 {
17     debug(1, "%s is called..\n", __func__);
18 }
```

4.4.1.10 f_visibility()

```
void f_visibility (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 27 번째 라인에서 정의되었습니다.

```
31 {
32     debug(1, "%s is called..\n", __func__);
33     return 0;
34 }
```

4.4.1.11 f_warn_unused_result()

```
int f_warn_unused_result (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 36 번째 라인에서 정의되었습니다.

```
39 {
40     debug(1, "%s is called..\n", __func__);
41 }
```

4.4.1.12 f_weakref()

```
void f_weakref (
    void )
```

Declaring_Attributes_of_Functions.c 파일의 42 번째 라인에서 정의되었습니다.

```
45 {
46     va_list ap;
47     va_start (ap, fmt);
48
49     vfprintf(stderr, fmt, ap);
50     va_end(ap);
51 }
```

4.5 src/function/Declaring_Attributes_of_Functions2.c 파일 참조

함수

- void [f_external](#) (void)

4.5.1 함수 문서화

4.5.1.1 f_external()

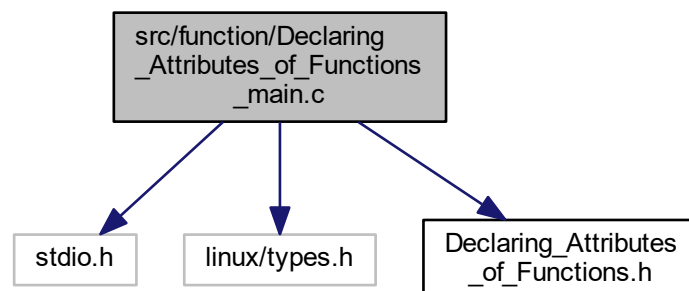
```
void f_external (
    void )
```

Declaring_Attributes_of_Functions2.c 파일의 2 번째 라인에서 정의되었습니다.

```
3 {
4     f_hidden();
5 }
```

4.6 src/function/Declaring_Attributes_of_Functions_main.c 파일 참조

```
#include <stdio.h>
#include <linux/types.h>
#include "Declaring_Attributes_of_Functions.h"
Declaring_Attributes_of_Functions_main.c에 대한 include 의존 그래프
```



함수

- static void `stf_weakref` (void)

변수

- void(* `p_func`)(void)

4.6.1 함수 문서화

4.6.1.1 stf_weakref()

```
static void stf_weakref (
    void ) [static]
```

Declaring_Attributes_of_Functions_main.c 파일의 7 번째 라인에서 정의되었습니다.

```
10 {
11     /* f_noreturn(); */
12     f_nonnull(NULL, NULL, 0);
13     p_func();
14     f_warn_unused_result();
15     stf_weakref();
16
17     return 0;
18 }
```

4.6.2 변수 문서화

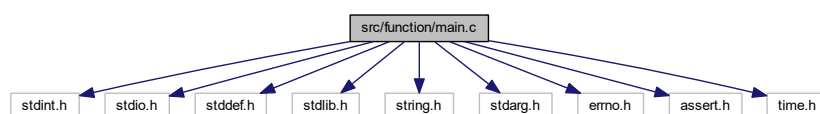
4.6.2.1 p_func

```
void(* p_func) (void)
```

4.7 src/function/main.c 파일 참조

```
#include <stdint.h>
#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <errno.h>
#include <assert.h>
#include <time.h>
```

main.c에 대한 include 의존 그래프



데이터 구조

- struct `inventory`

매크로

- #define `ARRAY_SIZE` 8
Array size
- #define `BUFF_SIZE` 256
Buffer size for character data

타입정의

- typedef struct `inventory` `INVENTORY`

함수

- void `operationError` (char *fmt,...)
- void `displInventory` (`INVENTORY` *ip)
- void `sleep` (int cnt)
- int `main` (int argc, char const *argv[])

4.7.1 상세한 설명

작성자

날짜

참고

4.7.2 매크로 문서화

4.7.2.1 `ARRAY_SIZE`

```
#define ARRAY_SIZE 8
```

Array size

main.c 파일의 22 번째 라인에서 정의되었습니다.

4.7.2.2 BUFF_SIZE

```
#define BUFF_SIZE 256
```

Buffer size for character data

main.c 파일의 23 번째 라인에서 정의되었습니다.

4.7.3 타입정의 문서화

4.7.3.1 INVENTORY

```
typedef struct inventory INVENTORY
```

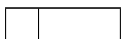
상품 재고 관리용 inventory 구조체 정의

4.7.4 함수 문서화

4.7.4.1 dispInventory()

```
void dispInventory (
    INVENTORY * ip )
```

매개변수



main.c 파일의 85 번째 라인에서 정의되었습니다.

```
85     {
86     printf("상품명 : %s\n", ip->name);
87     printf("상품번호 : %s\n", ip->number);
88     printf("재고 수량 : %d\n", ip->volume);
89     printf("매입 일수 : %d\n", ip->leadtime);
90 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



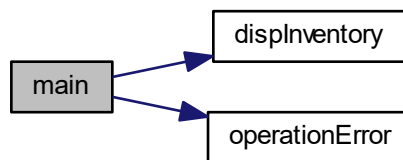
4.7.4.2 main()

```
int main (
    int argc,
    char const * argv[] )
```

main.c 파일의 106 번째 라인에서 정의되었습니다.

```
106             {
107
108     return 0;
109 }
```

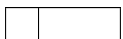
이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



4.7.4.3 operationError()

```
void operationError (
    char * fmt,
    ... )
```

매개변수



main.c 파일의 64 번째 라인에서 정의되었습니다.

```
65 {
66     char buff[BUFF_SIZE] = {'\0'};
67     va_list ap;
68
69     strcpy(buff, "ERROR: ");
70     va_start(ap, fmt);
71     vsprintf(buff + strlen(buff), fmt, ap);
72     va_end(ap);
73 }
```



```

74  if (puts(buff) == EOF) {
75      printf("Failed to write data to stdout!");
76  }
77
78  exit(1);
79 }

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



4.7.4.4 sleep()

```

void sleep (
    int cnt )

```

매개변수



main.c 파일의 96 번째 라인에서 정의되었습니다.

```

96      {
97  clock_t t = clock() * (1000 / CLOCKS_PER_SEC);
98  clock_t tterm = t + cnt;
99
100  while (t < tterm) {
101      t = clock() * (1000 / CLOCKS_PER_SEC);
102  }
103 }

```

4.8 src/pointer/main.c 파일 참조

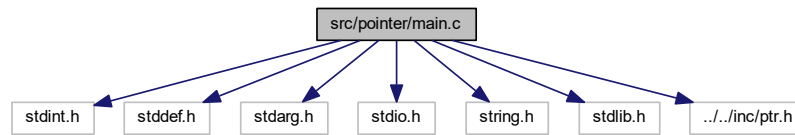
```

#include <stdint.h>
#include <stddef.h>
#include <stdarg.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

```

```
#include "../inc/ptr.h"
```

main.c에 대한 include 의존 그래프



매크로

- `#define ROW_MAX 5`
- `#define COL_MAX 10`

함수

- `int vaPrintf (char *fmt,...)`
- `int main (int argc, char const *argv[])`

변수

- `int **a = NULL`

4.8.1 매크로 문서화

4.8.1.1 COL_MAX

```
#define COL_MAX 10
```

main.c 파일의 10 번째 라인에서 정의되었습니다.

4.8.1.2 ROW_MAX

```
#define ROW_MAX 5
```

main.c 파일의 9 번째 라인에서 정의되었습니다.

4.8.2 함수 문서화

4.8.2.1 main()

```
int main (
    int arTgc,
    char const * argv[ ] )
```

main.c 파일의 27 번째 라인에서 정의되었습니다.

```
27                                     {
28
29     return 0;
30 }
```

4.8.2.2 vaPrintf()

```
int vaPrintf (
    char * fmt,
    ... )
```

main.c 파일의 14 번째 라인에서 정의되었습니다.

```
15 {
16     char buf[256] = {'\0',};
17     va_list ap;
18
19     strcpy(buf, "Error: ");
20     va_start(ap, fmt);
21     vsprintf(buf + strlen(buf), fmt, ap);
22     va_end(ap);
23
24     puts(buf);
25 }
```

4.8.3 변수 문서화

4.8.3.1 a

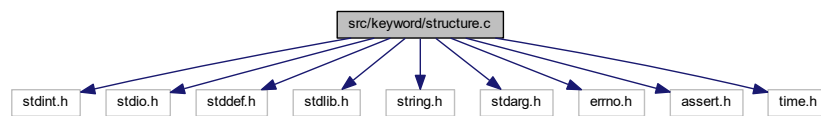
```
int** a = NULL
```

main.c 파일의 12 번째 라인에서 정의되었습니다.

4.9 src/keyword/structure.c 파일 참조

```
#include <stdint.h>
#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <errno.h>
#include <assert.h>
#include <time.h>
```

structure.c에 대한 include 의존 그래프



데이터 구조

- struct [_struct](#)

매크로

- #define [BUFF_SIZE](#) 256
Buffer size for character data

타입정의

- typedef struct [_struct](#) [STRUCT](#)

함수

- int [main](#) (int argc, char const *argv[])

4.9.1 상세한 설명

작성자

날짜

참고

4.9.2 매크로 문서화

4.9.2.1 BUFF_SIZE

```
#define BUFF_SIZE 256
```

Buffer size for character data

structure.c 파일의 22 번째 라인에서 정의되었습니다.

4.9.3 타입정의 문서화

4.9.3.1 STRUCT

```
typedef struct _struct STRUCT
```

4.9.4 함수 문서화

4.9.4.1 main()

```
int main (
    int argc,
    char const * argv[] )
```

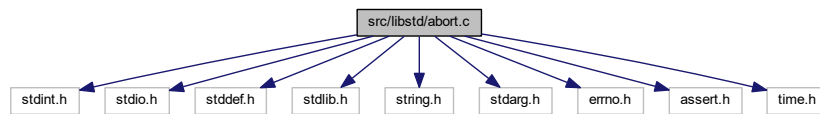
structure.c 파일의 54 번째 라인에서 정의되었습니다.

```
54                                     {
55
56     return 0;
57 }
```

4.10 src/libstd/abort.c 파일 참조

```
#include <stdint.h>
#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <errno.h>
#include <assert.h>
#include <time.h>
```

abort.c에 대한 include 의존 그래프



데이터 구조

- struct [_struct](#)

매크로

- #define [BUFF_SIZE](#) 256
Buffer size for character data

타입정의

- typedef struct [_struct](#) [STRUCT](#)

함수

- void [myAbort](#) (char *err)

4.10.1 상세한 설명

작성자

날짜

참고

4.10.2 매크로 문서화

4.10.2.1 BUFF_SIZE

```
#define BUFF_SIZE 256
```

Buffer size for character data

abort.c 파일의 22 번째 라인에서 정의되었습니다.

4.10.3 타입정의 문서화

4.10.3.1 STRUCT

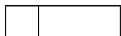
```
typedef struct _struct STRUCT
```

4.10.4 함수 문서화

4.10.4.1 myAbort()

```
void myAbort (
    char * err )
```

매개변수



abort.c 파일의 60 번째 라인에서 정의되었습니다.

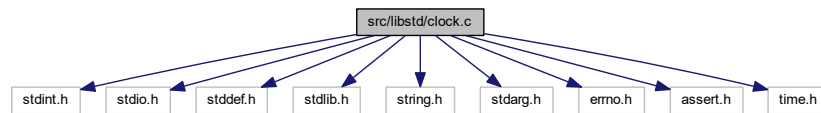
```
61 {
62     char buff[256] = "[ABORT]: ";
63     strncat(buff, err, strlen(err));
64     fprintf(stderr, buff, stderr);
65 }
```

4.11 src/libstd/clock.c 파일 참조

```
#include <stdint.h>
#include <stdio.h>
```

```
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <errno.h>
#include <assert.h>
#include <time.h>
```

clock.c에 대한 include 의존 그래프



데이터 구조

- struct [data_st](#)

매크로

- `#define BUFF_SIZE 256`
Buffer size for character data

타입정의

- typedef struct [data_st](#) DATA

함수

- void [myMemcpy](#) (void *dst, void *src, unsigned int size)

4.11.1 상세한 설명

작성자

날짜

참고

4.11.2 매크로 문서화

4.11.2.1 BUFF_SIZE

```
#define BUFF_SIZE 256
```

Buffer size for character data

clock.c 파일의 22 번째 라인에서 정의되었습니다.

4.11.3 타입정의 문서화

4.11.3.1 DATA

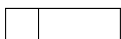
```
typedef struct data_st DATA
```

4.11.4 함수 문서화

4.11.4.1 myMemcpy()

```
void myMemcpy (
    void * dst,
    void * src,
    unsigned int size )
```

매개변수



clock.c 파일의 62 번째 라인에서 정의되었습니다.

```
63 {
64     char *s = src;
65     char *d = dst;
66     size_t pad = size % sizeof(int);
67
68     for (size_t i = 0; i < size / sizeof(int); i++) {
69         *(int *)d = *(int *)s;
70         s += 4;
71         d += 4;
72     }
73
74     for (size_t i = 0; i < pad; i++) {
```

```

75     *d = *s;
76     d++;
77     s++;
78 }
79 }

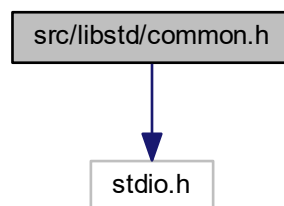
```

4.12 src/libstd/common.h 파일 참조

```

#include <stdio.h>
common.h에 대한 include 의존 그래프

```

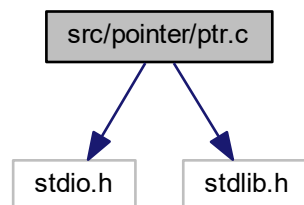


4.13 src/pointer/ptr.c 파일 참조

```

#include <stdio.h>
#include <stdlib.h>
ptr.c에 대한 include 의존 그래프

```



매크로

- `#define STACK_SIZE 10`

타입정의

- typedef int `stack_t`

함수

- void `push` (int data)
- int `pop` (void)

변수

- `stack_t stack` [`STACK_SIZE`]
- unsigned int `top`

4.13.1 매크로 문서화

4.13.1.1 STACK_SIZE

```
#define STACK_SIZE 10
```

ptr.c 파일의 4 번째 라인에서 정의되었습니다.

4.13.2 타입정의 문서화

4.13.2.1 stack_t

```
typedef int stack_t
```

ptr.c 파일의 5 번째 라인에서 정의되었습니다.

4.13.3 함수 문서화

4.13.3.1 pop()

```
int pop (
    void )
```

ptr.c 파일의 16 번째 라인에서 정의되었습니다.

```
17 {
18     if (top == 0)
19         exit(EXIT_FAILURE);
20
21     return stack[--top];
22 }
```

4.13.3.2 push()

```
void push (
    int data )
```

ptr.c 파일의 9 번째 라인에서 정의되었습니다.

```
9     {
10     if (top >= STACK_SIZE)
11         exit(EXIT_FAILURE);
12
13     stack[++top] = data;
14 }
```

4.13.4 변수 문서화

4.13.4.1 stack

```
stack_t stack[STACK_SIZE]
```

ptr.c 파일의 6 번째 라인에서 정의되었습니다.

4.13.4.2 top

```
unsigned int top
```

ptr.c 파일의 7 번째 라인에서 정의되었습니다.

4.14 src/pointer/ptr.h 파일 참조

Index

- `__func_nonnull`
 - Declaring_Attributes_of_Functions.h, [17](#)
 - `__func_noreturn`
 - Declaring_Attributes_of_Functions.c, [13](#)
 - Declaring_Attributes_of_Functions.h, [17](#)
 - `__func_visibility`
 - Declaring_Attributes_of_Functions.c, [14](#)
 - Declaring_Attributes_of_Functions.h, [17](#)
 - `__func_warn_unused_result`
 - Declaring_Attributes_of_Functions.h, [18](#)
 - `__func_weakref`
 - Declaring_Attributes_of_Functions.h, [18](#)
- `_struct`, [5](#)
 - i, [5](#)
- a
 - pointer/main.c, [27](#)
- ARRAY_SIZE
 - function/main.c, [22](#)
- abort.c
 - BUFF_SIZE, [31](#)
 - myAbort, [31](#)
 - STRUCT, [31](#)
- age
 - data_st, [6](#)
- alias
 - Declaring_Attributes_of_Functions.c, [14](#)
- BUFF_SIZE
 - abort.c, [31](#)
 - clock.c, [33](#)
 - function/base.c, [10](#)
 - function/main.c, [22](#)
 - libstd/base.c, [12](#)
 - structure.c, [29](#)
- COL_MAX
 - pointer/main.c, [26](#)
- ch
 - data_st, [7](#)
- clock.c
 - BUFF_SIZE, [33](#)
 - DATA, [33](#)
 - myMemcpy, [33](#)
- DATA
 - clock.c, [33](#)
- data_st, [6](#)
 - age, [6](#)
 - ch, [7](#)
- name, [7](#)
- debug
 - Declaring_Attributes_of_Functions.c, [14](#)
 - Declaring_Attributes_of_Functions.h, [18](#)
- Declaring_Attributes_of_Functions.c
 - `__func_noreturn`, [13](#)
 - `__func_visibility`, [14](#)
 - alias, [14](#)
 - debug, [14](#)
 - f_nonnull, [15](#)
 - f_noreturn, [15](#)
 - f_visibility, [15](#)
 - f_warn_unused_result, [16](#)
 - f_weakref, [16](#)
- Declaring_Attributes_of_Functions.h
 - `__func_nonnull`, [17](#)
 - `__func_noreturn`, [17](#)
 - `__func_visibility`, [17](#)
 - `__func_warn_unused_result`, [18](#)
 - `__func_weakref`, [18](#)
 - debug, [18](#)
 - error, [18](#)
 - f_nonnull, [19](#)
 - f_noreturn, [19](#)
 - f_visibility, [19](#)
 - f_warn_unused_result, [19](#)
 - f_weakref, [19](#)
- Declaring_Attributes_of_Functions2.c
 - f_external, [20](#)
- Declaring_Attributes_of_Functions_main.c
 - p_func, [21](#)
 - stf_weakref, [21](#)
- displInventory
 - function/main.c, [23](#)
- error
 - Declaring_Attributes_of_Functions.h, [18](#)
- f_external
 - Declaring_Attributes_of_Functions2.c, [20](#)
- f_nonnull
 - Declaring_Attributes_of_Functions.c, [15](#)
 - Declaring_Attributes_of_Functions.h, [19](#)
- f_noreturn
 - Declaring_Attributes_of_Functions.c, [15](#)
 - Declaring_Attributes_of_Functions.h, [19](#)
- f_visibility
 - Declaring_Attributes_of_Functions.c, [15](#)
 - Declaring_Attributes_of_Functions.h, [19](#)
- f_warn_unused_result

- Declaring_Attributes_of_Functions.c, 16
- Declaring_Attributes_of_Functions.h, 19
- f_weakref
 - Declaring_Attributes_of_Functions.c, 16
 - Declaring_Attributes_of_Functions.h, 19
- function/base.c
 - BUFF_SIZE, 10
 - main, 10
 - STRUCT, 10
- function/main.c
 - ARRAY_SIZE, 22
 - BUFF_SIZE, 22
 - displInventory, 23
 - INVENTORY, 23
 - main, 24
 - operationError, 24
 - sleep, 25
- i
 - _struct, 5
- INVENTORY
 - function/main.c, 23
- inventory, 7
 - leadtime, 8
 - name, 8
 - number, 8
 - volume, 8
- leadtime
 - inventory, 8
- libstd/base.c
 - BUFF_SIZE, 12
 - main, 12
 - STRUCT, 12
- main
 - function/base.c, 10
 - function/main.c, 24
 - libstd/base.c, 12
 - pointer/main.c, 26
 - structure.c, 29
- myAbort
 - abort.c, 31
- myMemcpy
 - clock.c, 33
- name
 - data_st, 7
 - inventory, 8
- number
 - inventory, 8
- operationError
 - function/main.c, 24
- p_func
 - Declaring_Attributes_of_Functions_main.c, 21
- pointer/main.c
 - a, 27
 - COL_MAX, 26
 - main, 26
 - ROW_MAX, 26
 - vaPrintf, 27
- ptr.c, 35
 - pop, 35
 - push, 36
 - STACK_SIZE, 35
 - stack, 36
 - stack_t, 35
 - top, 36
- push
 - ptr.c, 36
- ROW_MAX
 - pointer/main.c, 26
- STACK_SIZE
 - ptr.c, 35
- STRUCT
 - abort.c, 31
 - function/base.c, 10
 - libstd/base.c, 12
 - structure.c, 29
- sleep
 - function/main.c, 25
- src/function/Declaring_Attributes_of_Functions.c, 13
- src/function/Declaring_Attributes_of_Functions.h, 16
- src/function/Declaring_Attributes_of_Functions2.c, 20
- src/function/Declaring_Attributes_of_Functions_main.c, 20
- src/function/base.c, 9
- src/function/main.c, 21
- src/keyword/structure.c, 28
- src/libstd/abort.c, 30
- src/libstd/base.c, 11
- src/libstd/clock.c, 31
- src/libstd/common.h, 34
- src/pointer/main.c, 25
- src/pointer/ptr.c, 34
- src/pointer/ptr.h, 36
- stack
 - ptr.c, 36
- stack_t
 - ptr.c, 35
- stf_weakref
 - Declaring_Attributes_of_Functions_main.c, 21
- structure.c
 - BUFF_SIZE, 29
 - main, 29
 - STRUCT, 29
- top
 - ptr.c, 36
- vaPrintf
 - pointer/main.c, 27
- volume
 - inventory, 8