libft复习卡片

### **ft\_atoi**

* 作用：把 C 字符串转换为 int（类似标准库 atoi）。
* 参数：str — const char \*。
* 返回值：int。

### **ft\_bzero**

* 作用：将内存前 n 字节置零（bzero 行为）。
* 参数：s — void \*；n — size\_t。
* 返回值：void。

### **ft\_calloc**

* 作用：分配 nmemb \* size 字节并清零（类似 calloc）。
* 参数：nmemb — size\_t；size — size\_t。
* 返回值：void \*（分配指针，失败返回 NULL）。

### **ft\_isalnum**

* 作用：判断字符是否为字母或数字（ctype 风格）。
* 参数：c — int。
* 返回值：int（0 表示 false，非 0 表示 true）。

### **ft\_isalpha**

* 作用：判断字符是否为字母。
* 参数：c — int。
* 返回值：int。

### **ft\_isascii**

* 作用：判断是否为 ASCII 字符（0–127）。
* 参数：c — int。
* 返回值：int。

### **ft\_isdigit**

* 作用：判断字符是否为数字（'0'..'9'）。
* 参数：c — int。
* 返回值：int。

### **ft\_isprint**

* 作用：判断字符是否可打印（含空格）。
* 参数：c — int。
* 返回值：int。

### **count\_num**

* 作用：辅助函数——计算整数的数字位数（常用于 itoa）。
* 参数：n — int。
* 返回值：int（数字位数）。

### **ft\_itoa**

* 作用：将 int 转为以 '\0' 结尾的字符串（动态分配）。
* 参数：n — int。
* 返回值：char \*（需 free）。

### **ft\_lstadd\_back**

* 作用：把节点追加到链表尾（单链表）。
* 参数：lst — t\_list \*\*（指向头指针）；new — t\_list \*（新节点）。
* 返回值：void。

### **ft\_lstadd\_front**

* 作用：把节点插入链表头。
* 参数：lst — t\_list \*\*；new — t\_list \*。
* 返回值：void。

### **ft\_lstclear**

* 作用：删除并释放整个链表（使用 del 对内容释放）。
* 参数：lst — t\_list \*\*；del — void (\*)(void \*)。
* 返回值：void。

### **ft\_lstdelone**

* 作用：删除并释放单个节点（使用 del 释放内容）。
* 参数：lst — t\_list \*；del — void (\*)(void \*)。
* 返回值：void。

### **ft\_lstiter**

* 作用：对链表每个节点的 content 调用函数 f。
* 参数：lst — t\_list \*；f — void (\*)(void \*)。
* 返回值：void。

### **ft\_lstlast**

* 作用：返回链表最后一个节点指针。
* 参数：lst — t\_list \*。
* 返回值：t\_list \*（最后节点或 NULL）。

### **ft\_lstmap**

* 作用：对链表每个节点应用 f 并生成新链表（若失败调用 del 清理）。
* 参数：lst — t\_list \*；f — void \*(\*)(void \*)；del — void (\*)(void \*)。
* 返回值：t\_list \*（新链表头或 NULL）。

### **ft\_lstnew**

* 作用：创建并返回新链表节点，节点 content 设为传入值。
* 参数：content — void \*。
* 返回值：t\_list \*（新节点或 NULL）。

### **ft\_lstsize**

* 作用：返回链表节点数（长度）。
* 参数：lst — t\_list \*。
* 返回值：int。

### **ft\_memchr**

* 作用：在内存块中查找字节 c（类似 memchr）。
* 参数：s — const void \*；c — int；n — size\_t。
* 返回值：void \*（指向首次出现处或 NULL）。

### **ft\_memcmp**

* 作用：比较两块内存的前 n 字节（类似 memcmp）。
* 参数：s1 — const void \*；s2 — const void \*；n — size\_t。
* 返回值：int（<0 / 0 / >0）。

### **ft\_memcpy**

* 作用：复制内存（不保证处理重叠）。
* 参数：dst — void \*；src — const void \*；n — size\_t。
* 返回值：void \*（dst）。

### **memmove\_copy**

* 作用：内部 helper，处理 memmove 的复制逻辑（通常处理向后拷贝或向前拷贝）。
* 参数：dst — void \*；src — const void \*；n — size\_t。
* 返回值：void \*（dst）。

### **ft\_memmove**

* 作用：安全复制内存，支持重叠区域（类似 memmove）。
* 参数：dst — void \*；src — const void \*；len — size\_t。
* 返回值：void \*（dst）。

### **ft\_memset**

* 作用：用指定字节填充内存（类似 memset）。
* 参数：b — void \*；c — int；len — size\_t。
* 返回值：void \*（b）。

### **dispatch\_format**

* 作用：printf 内部分发格式解析并调用相应打印函数（实现细节）。
* 参数：（实现依赖）通常有格式字符、va\_list、输出缓冲等。
* 返回值：通常 int（输出字符数或错误码）。

### **ft\_printf**

* 作用：格式化输出到 stdout（printf 风格实现）。
* 参数：format — const char \*，可变参数 ...。
* 返回值：int（写入字符总数或错误）。

### **print\_char**

* 作用：printf helper——打印字符格式 %c。
* 参数：通常 va\_list \* 或 int c（实现不同）。
* 返回值：int（写入字符数）。

### **print\_decimal**

* 作用：printf helper——打印带符号十进制整数 %d / %i。
* 参数：va\_list \* 或 int。
* 返回值：int（写入字符数）。

### **print\_hex**

* 作用：printf helper——打印十六进制整数 %x / %X。
* 参数：va\_list \* 或 unsigned int。
* 返回值：int（写入字符数）。

### **print\_percent**

* 作用：处理 %% —— 输出字面 %。
* 参数：通常无（或格式上下文）。
* 返回值：int（写入字符数，通常 1）。

### **print\_pointer**

* 作用：printf helper——打印指针地址 %p。
* 参数：va\_list \* 或 void \*。
* 返回值：int（写入字符数）。

### **print\_string**

* 作用：printf helper——打印字符串 %s。
* 参数：va\_list \* 或 char \*。
* 返回值：int（写入字符数）。

### **print\_unsigned**

* 作用：printf helper——打印无符号整数 %u。
* 参数：va\_list \* 或 unsigned int。
* 返回值：int（写入字符数）。

### **count\_digit**

* 作用：辅助函数，计算无符号数在某个进制下的位数。
* 参数：n — unsigned long；base — int。
* 返回值：int（位数）。

### **utoa\_base**

* 作用：将无符号整数按给定进制转换为字符串（动态分配）。
* 参数：n — unsigned long；base — int。
* 返回值：char \*（需 free）。

### **ft\_putchar**

* 作用：向 stdout 输出单字符（实现可能返回写入字节数）。
* 参数：c — char。
* 返回值：int 或 void（实现不同；常为写入字节数）。

### **ft\_putchar\_fd**

* 作用：向指定文件描述符写入单个字符。
* 参数：c — char；fd — int。
* 返回值：void。

### **ft\_putendl\_fd**

* 作用：向 fd 写入字符串并追加换行。
* 参数：s — char \*；fd — int。
* 返回值：void。

### **ft\_putnbr\_fd**

* 作用：向 fd 写入整数的 ASCII 表示。
* 参数：n — int；fd — int。
* 返回值：void。

### **ft\_putstr\_fd**

* 作用：向 fd 写入字符串（不含换行）。
* 参数：s — char \*；fd — int。
* 返回值：void。

### **ft\_split**

* 作用：按单字符分隔符分割字符串，返回以 NULL 结尾的字符串数组（动态分配）。
* 参数：s — char \*；c — char（分隔符）。
* 返回值：char \*\*（需逐项 free）。

### **ft\_strchr**

* 作用：查找字符串中首次出现的字符（类似 strchr）。
* 参数：s — const char \*；c — int。
* 返回值：char \*（位置或 NULL）。

### **ft\_strdup**

* 作用：复制字符串并返回新分配的副本（strdup）。
* 参数：s1 — const char \*。
* 返回值：char \*（需 free）。

### **ft\_striteri**

* 作用：对字符串的每个字符按索引调用回调 f（原地修改）。
* 参数：s — char \*；f — void (\*)(unsigned int, char \*)。
* 返回值：void。

### **ft\_strjoin**

* 作用：拼接两个字符串并返回新分配的结果。
* 参数：s1 — char \*；s2 — char \*。
* 返回值：char \*（需 free）。

### **ft\_strlcat**

* 作用：将 src 追加到 dst，保证总长度不超过 dstsize（strlcat 行为）。
* 参数：dst — char \*；src — const char \*；dstsize — size\_t。
* 返回值：size\_t（尝试拼接之后的字符串长度，便于检测截断）。

### **ft\_strlcpy**

* 作用：拷贝字符串到 dst 并保证 NUL 结尾（strlcpy 行为）。
* 参数：dst — char \*；src — const char \*；dstsize — size\_t。
* 返回值：size\_t（源字符串长度）。

### **ft\_strlen**

* 作用：计算 C 字符串长度（不含终结符）。
* 参数：s — const char \*。
* 返回值：size\_t。

### **ft\_strmapi**

* 作用：对字符串每个字符调用函数 f 并返回新字符串（不修改原串）。
* 参数：s — char const \*；f — char (\*)(unsigned int, char)。
* 返回值：char \*（需 free）。

### **ft\_strncmp**

* 作用：比较两个字符串的前 n 个字符（类似 strncmp）。
* 参数：s1 — const char \*；s2 — const char \*；n — size\_t。
* 返回值：int。

### **ft\_strnstr**

* 作用：在最多 len 个字符范围内查找子串（类似 strnstr）。
* 参数：haystack — const char \*；needle — const char \*；len — size\_t。
* 返回值：char \*（匹配处或 NULL）。

### **ft\_strrchr**

* 作用：查找字符串中最后一次出现的字符（类似 strrchr）。
* 参数：s — const char \*；c — int。
* 返回值：char \*（位置或 NULL）。

### **ft\_strtrim**

* 作用：去掉字符串开头与末尾位于 set 中的字符，返回新字符串。
* 参数：s1 — char const \*；set — char const \*。
* 返回值：char \*（需 free）。

### **ft\_substr**

* 作用：从 s 的 start 位置截取最多 len 个字符作为子串（分配返回）。
* 参数：s — char const \*；start — unsigned int；len — size\_t。
* 返回值：char \*（需 free）。

### **ft\_tolower**

* 作用：把字母字符转换为小写（tolower 行为）。
* 参数：c — int。
* 返回值：int（转换后的字符或原值）。

### **ft\_toupper**

* 作用：把字母字符转换为大写（toupper 行为）。
* 参数：c — int。
* 返回值：int。

### **join\_free**

* 作用：拼接两个字符串并根据 mode 释放其中一个或两个（GNL/实用 helper）。
* 参数：s1 — char \*；s2 — char \*；mode — int（决定释放策略）。
* 返回值：char \*（拼接结果，需 free）。

### **read\_save**

* 作用：从 fd 读取数据并追加保存到缓冲（get\_next\_line helper）。
* 参数：fd — int；save — char \*（现有缓冲）。
* 返回值：char \*（更新的保存缓冲或 NULL）。

### **get\_the\_line**

* 作用：从保存的缓冲中提取下一行（含换行），作为返回值（GNL helper）。
* 参数：save — char \*。
* 返回值：char \*（分配的新行）。

### **refresh\_backup**

* 作用：在提取行后刷新保存缓冲，移除已返回部分（GNL helper）。
* 参数：save — char \*。
* 返回值：char \*（更新的保存缓冲或 NULL）。

### **get\_next\_line**

* 作用：按行读取文件描述符，返回下一行（含换行，最后无换行返回剩余内容）。
* 参数：fd — int。
* 返回值：char \*（分配字符串或 NULL 表示结束/错误）。

### **gnl\_strlen**

* 作用：GNL 内部用的字符串长度函数。
* 参数：s — char \*。
* 返回值：size\_t。

### **gnl\_strchr**

* 作用：GNL 的查找字符 helper（寻找 \n 等）。
* 参数：s — char \*；c — int。
* 返回值：char \*（位置或 NULL）。

### **gnl\_strjoin**

* 作用：GNL 专用的字符串拼接（内部使用，处理 NULL 等）。
* 参数：s1 — char \*；s2 — char \*。
* 返回值：char \*（需 free）。

### **gnl\_strdup**

* 作用：GNL 专用字符复制（内部 helper）。
* 参数：s1 — char \*。
* 返回值：char \*。

### **gnl\_substr**

* 作用：GNL 专用的子串提取 helper。
* 参数：s — char \*；start — unsigned int；len — size\_t。
* 返回值：char \*。