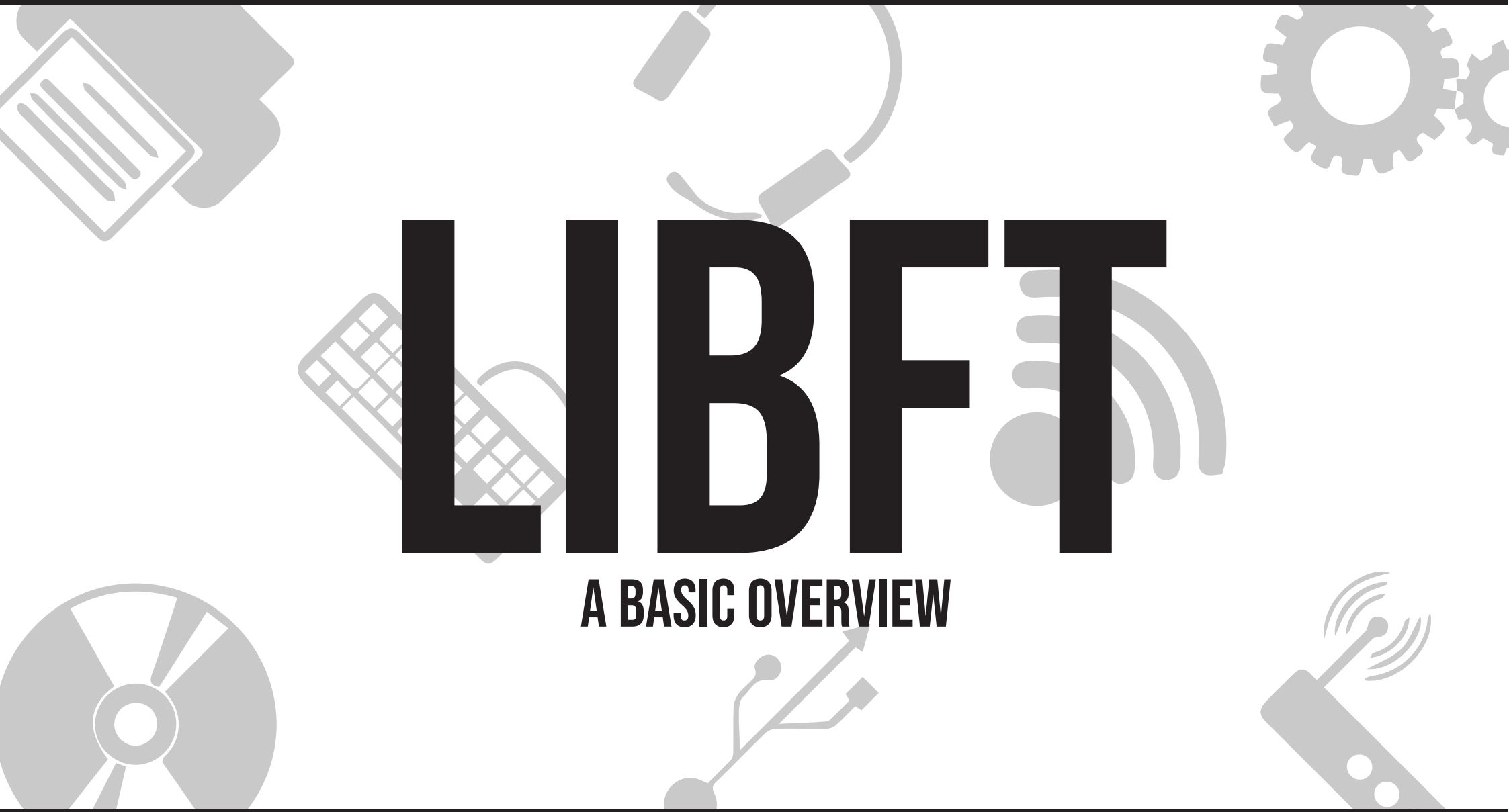
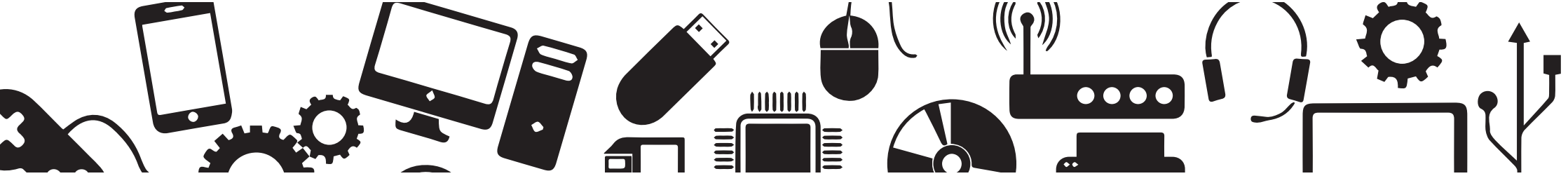


LIBFT

A BASIC OVERVIEW

PRESENTED BY: BNKOSI, MMODISAD, KMBUKUTS, TMOKOENA

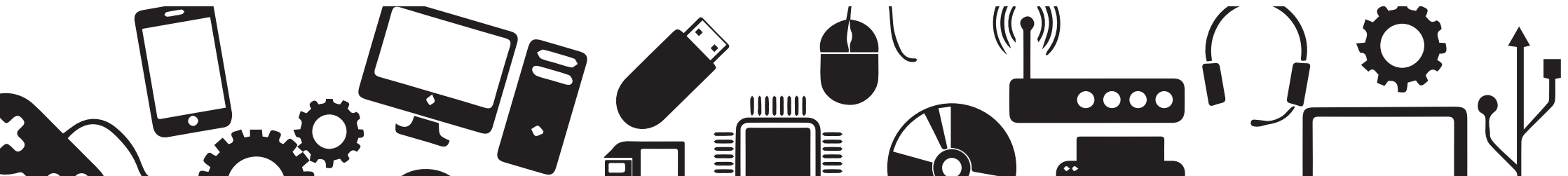


OBJECTIVE:

LIBFT PROJECT AIMS TO MAKE THE STUDENT REMAKE /RE-WRITE STANDARD C FUNCTIONS
-> UNDERSTAND THEM -> UNDERSTAND HOW TO USE THEM.

LIBFT DIVIDED INTO 2 PARTS

1. RECREATING EXISTING STANDARD FUNCS (THEY HAVE A MAN)
2. MAKE FUNCS THAT ARE EITHER NOT INCLUDED IN LIBC OR THEY ARE INCLUDEDD IN A DIFFERENT FORM



HI_MY_NAME_IS_BOB_

- **STRLEN** MEASURES LENGTH OF A STRING, RETURNS INTEGER.
- **STRDUP** DUPLICTES STRING, RETURNS POINTER TO NULL TERMINATED STRING. DYNAMIC
- **STRCPY** COPIES STRING, RETURNS POINTER TO THE MEMORY POINTED TO BY 'DEST'. STATIC
- **STRCAT** APPENDS/ADDS A COPY OF ONE STRING TO ANOTHER STRING.
- **STRNCPY** COPIES 'N' AMOUNT OF CHARACTERS FROM 1ST STRING, RETURNS POINTER TO THE MEMORY POINTED TO BY 'DEST'.
- **STRNCAT** APPENDS/ADDS A COPY OF ONE STRING TO ANOTHER STRING FOR NO MORE THAN 'N' BYTES (SIZE DEPENDANT)
- **STRCHR** FINDS FIRST OCCURRENCE OF A CHAR AND RETURNS A POINTER TO IT. NULL IF IT AINT THERE
- **STRLCAT** APPENDS/ADDS A COPY OF ONE STRING TO ANOTHER STRING - SAFER THAN STRNCAT
- **STRRCHR** FINDS LAST OCCURRENCE OF A CHAR AND RETURNS A POINTER TO IT. NULL IF IT AINT THERE
- **STRSTR** FINDS A (SUB)STRING IN A STRING.
- **STRNSTR** FINDS A (SUB)STRING IN A STRING WHERE NO MORE THAN N CHARACTERS ARE SEARCHED.
- **STRCMP** COMPARES 2 STRINGS AND RETURNS AN INTERGER INDICATING RELATIONSHIP BETWEEN THE TWO STRINGS. DIFFERENCE OF 2
- **STRNCMP** COMPARES 2 STRINGS AND RETURNS DIFFERENCE BETWEEN 2 FOR LENGTH OF N.

B

- **PUTCHAR** WRITES 1 CHARACTER TO STD OUTPUT
- **PUTSTR** WRITES MULTIPLE CHARACTERS TO STD OUTPUT
- **STRJOIN** PRETTY MUCH LIKE STRCAT EXCEPT IT RETURNS A NULL IF ALLOCATION FAILS
- **STRNEW** MAKES A FRESH STR, EACH CHARACTER OF THE STRING IS INITIALIZED TO '\0'
- **STRTRIM** REMOVES SPACES AT START AND END OF A STRING
- **PUTNBR** WRITES INTERGERS AS CHARS ONTO STD OUTPUT
- **STRMAP** APPLY'S A FUNCTION TO EACH CHARACTER IN A STRING. CREATES NEW STR
- **STRMAPI** APPLY'S A FUNCTION TO EACH CHARACTER IN A STRING BY GIVING ITS INDEX AS FIRST ARGUMENT.
- **STRCLR** SETS ALL CHARS IN A STRING TO NULL
- **PUTCHAR_FD** WRITES A SINGLE CHAR TO A DEFINED FILE DESCRIPTER.
- **STRSUB** MALLOCS A FRESH SUBSTRING FROM A STRING.
- **STRDEL** UNLIKE STRING CLEAR, IT DELETES A STRING - USES FREE (FREE'S MEMORY)
- **ITOA** CONVERTS INTERGER /S TO STRING.

M

LET_MY_MEMORY_GO!

- **STRITER** APPLIES A FUNCTION TO EACH CHAR IN A STRING. DOESNT MAKE FRESH STR
- **STREQU** COMPARES 2 STRINGS AND RETURNS 1 IF THEY ARE THE SAME AND 0 IF NOT.
- **STRNEQU** COMPARES 2 STRINGS TO 'N' AMOUNT OF BYTES AND RETURNS 1 IF THEY ARE THE SAME AND 0 IF NOT.
- **STRSPLIT** SEPERATES A STRING BY A DELIMITER INTO AN ARRAY OF NULL TERM'D STRINGS.
- **MEMALLOC** PRETTY MUCH MALLOC
- **MEMDEL** FREE'S ALLOCATED MEMORY
- **MEMCPY** COPIES BYTES FROM SOURCE TO DESTINATION
- **MEMCCPY** COPIES BYTES FROM SOURCE TO DESTINATION TILL 'C' IS REACHED
- **MEMMOVE** IT DOES THE SAME AS MEMCPY, BUT SUPPORTS OVERLAPPING.
- **MEMCHR** FINDS FIRST OCCURENCE OF CHAR 'C' IN FIRST N BYTES OF THE STRING POINTED TO.
- **MEMCMP** COMPARES THE DIFFERENCE BETWEEN BYTES OF 2 MEMORY ADDRESSES
- **MEMSET** SETS EACH BYTE OF MEMORY ADDRESS TO NULL.

$500 BONUS ROUND!!!

- DATA STRUCTURES
- LINKED LISTS
- MEMORY IS ALLOCATED DYNAMICALLY
- SIZE IS NOT FIXED
- DATA IS STORED IN DIFFERENT LOCATION IN COMPUTER MEMORY
- IT HAS AT LEAST 2 PARTS:
- SINGLY LINKED LIST
- DOUBLY LINKED LIST
- CIRCULAR LINKED LIST
- WHEN TO USE LINKED LIST??

CONSTANTLY INSERTING OR DELETING SOMETHING FROM LIST
UNKNOWN NUMBER OF ITEMS THAT WILL BE INSERTED
NO NEED FOR RANDOM ACCESS

K

THANK YOU

PRESENTED BY: BNKOSI, MMODISAD, KMBUKUTS, TMOKOENA