

Design Document Assignment 3

Purpose:

The purpose of this program is to perform a second pass of the linker. The objective is to create a complete linker by using the resulting table from the first pass of the linker (Assignment 2). An xme file is to be created based on the filename of the mainfile. In this pass, we will primarily be editing the L3 directives and updating the address and checksums of the S1 records accordingly. Checks for each of the L0 records will occur against the L1 symbol table created in the first pass; if found, the L3 records will be decoded and stored in an S1 record file within the xme output.

Algorithm:

Blue = A3 design

Void Main Function (No Parameters):

AT END OF FIRST PASS CODE

PASS first filename stored in filename array to create_xme FUNCTION

Void Create_XME Function(Parameters: filename)

SPLIT the input filename USING '.' AS A DELIMITER

APPEND '.xme' to the input filename without the extension

WRITE a '.xme' FILE WITH provided filename + '.xme' extension

Void Parse File Function (Parameters: filename, file counter, max number of files)

DECLARE last address in global

OPEN file

STORE contents of file in CHAR ARRAY wholefile

IF file counter==0

SET the last_address variable to 0

END IF

DECLARE INT interim relocation address.

DECLARE INT publics variable

DECLARE INT module starting address found as 0

LOOP through CHAR ARRAY wholefile

READ THROUGH IT LINE BY LINE

LOOP THROUGH EACH CHARACTER OF THE LINE

IF the first character is L && second character IS 1

SPLIT string array with " " delimiter

STORE the MIDDLE STRING AS THE symbol

STORE THE END STRING AS THE offset

SUM interim relocation address, offset, and last address
STORE in offsetted address variable

STORE symbol AND offsetted address in STRUCT TABLE
PRINT symbol, offset, and offsetted address
INCREMENT publics

END IF

ELSE IF the first character is L && the second character is 0
STORE CHARACTERS 3 – LAST CHARACTER
APPEND TO L3 character array

ELSE IF the first character is L && the second character is 3
SPLIT string array with “ ” delimiter
STORE the MIDDLE string as the I0_symbol
STORE the END string as the I0_index
STORE I0_symbol & I0_index in array of structs for L0

ELSE IF the first character is S && second character is 1
IF THE module's starting address found IS 0
 LOOP through S1
 STORE the 5th to 8th character in
 mod_st_address_arr
 CONVERT to int value mod_starting_address
 SET mod_starting_address_found to 1
 END LOOP
END IF

LOOP THROUGH S1
 STORE the 5th to 8th character in mod_s1_address_arr
 CONVERT using STRTOL to int value s1_starting_address
 STORE the 2nd and 3rd character as character array
 length_arr
 CONVERT length_arr to INTEGER length
 CALCULATE num_bytes = length-3 bytes (address and
 checksum)
 STORE interim_ra AS starting_address + num_bytes
 ADD interim_ra to last_address
 STORE as relocation_address
 REPLACE the 5th to 8th character with relocation address
 STORE chksum AS LAST TWO values of S1 array
 STORE updated_chksum AS ~(~(chksum)+hextring of
 characters 5 & 6 (low byte of relocation address)+
 hextring of characters 7 & 8)

```
                REPLACE checksum WITH updated_checksum on S1
                CALL append_to_xme FUNCTION and pass modified S1
                value to it
            END LOOP
        END IF
    IF publics ==0
        PRINT "No Publics Found."
    END IF

    STORE last_address as interim_ra+last_address. (This will be stored globally to be accessible by
    the next file.)

    RESET interim_ra to 0

    IF the current file is the last file
        PRINT symbol table WITH symbol && address
    END IF
```

PASS L1 ARRAY FROM FIRST PASS, L3 ARRAY, AND L0 ARRAY TO HandleL3L0 FUNCTION

END FUNCTION

AppendToXME Function(char array)

APPEND parameter to created FILE

HandleL3L1 Function (Params: L0 array, L1 array, L3 array):

```
FOR EACH L0 record:
    LOOP THROUGH L1 array
        CHECK for L0 SYMBOL IN L1 ARRAY
        IF NONE FOUND
            PRINT DIAGNOSTIC
        END IF
        ELSE
            EXTRACT THE RELOCATION ADDRESS FROM L1 array
    END LOOP
END FOR
FOR EACH L3 record in L3 array
    EXTRACT LAST 3 CHARACTERS FOR INDEX LOCATION
    EXTRACT 4 CHARACTERS AFTER L3 AS ADDRESS
    EXTRACT 5TH CHARACTER AS BL OR BRA INDICATOR
    MAP INDEX LOCATION ON L0 array and EXTRACT target_address

    STORE OFFSET AS OFFSET=target_address-(address+2)
```

```
    RIGHT SHIFT BY 1 AND STORE IN variable ENCODED
    STORE last two characters as LOWENCODED
    STORE first two characters as HIGHENCODED
    STORE LENGTH AS 5
    SUM LOWENCODED, HIGH ENCODED, LENGTH, AND ADDRESS
    COMPLEMENT ABOVE SUM AND STORE AS CHKSUM
        STORE S1 RECORD AS
        S1+LENGTH+ADDRESS+LOWENCODED+HIGHENCODED+CHKSUM
    CALL append_to_xme FUNCTION and pass S1 value to it
END FOR
```

Data Dictionary:

S1 = LENGTH+ADDRESS+ENCODED +CHKSUM

ENCODED = Instruction+ LOWENCODED+HIGHENCODED+

OFFSET = TARGET-(Address+2)

updated_chksum = ~(~(chksum)+low byte relocation address + high byte relocation address))