

## ***heap/stack size optimization***

### CONTENTS

- [Abstract](#)
- [Example of the issue](#)
- [global variables refactor](#)
  - [packing](#)
  - [cutting](#)
  - [if outside of struct](#)
  - [Pros & cons](#)
  - [limitation of the compiler Flacc](#)
- [local variables refactor](#)
  - [a simple example in outline](#)
  - [another example](#)
  - [Pros & cons](#)
- [design refactor](#)

*Xu YangChun Aug 592019*

# Abstract

During the maintenance phase, you may met the memory run out issue. If unfortunately, it is legacy code's issue, you have to find out where and how to free some space. This short article take scheduler thread(a FO of MAC layer) as example to describe the methods can be employed.

# Example of the issue

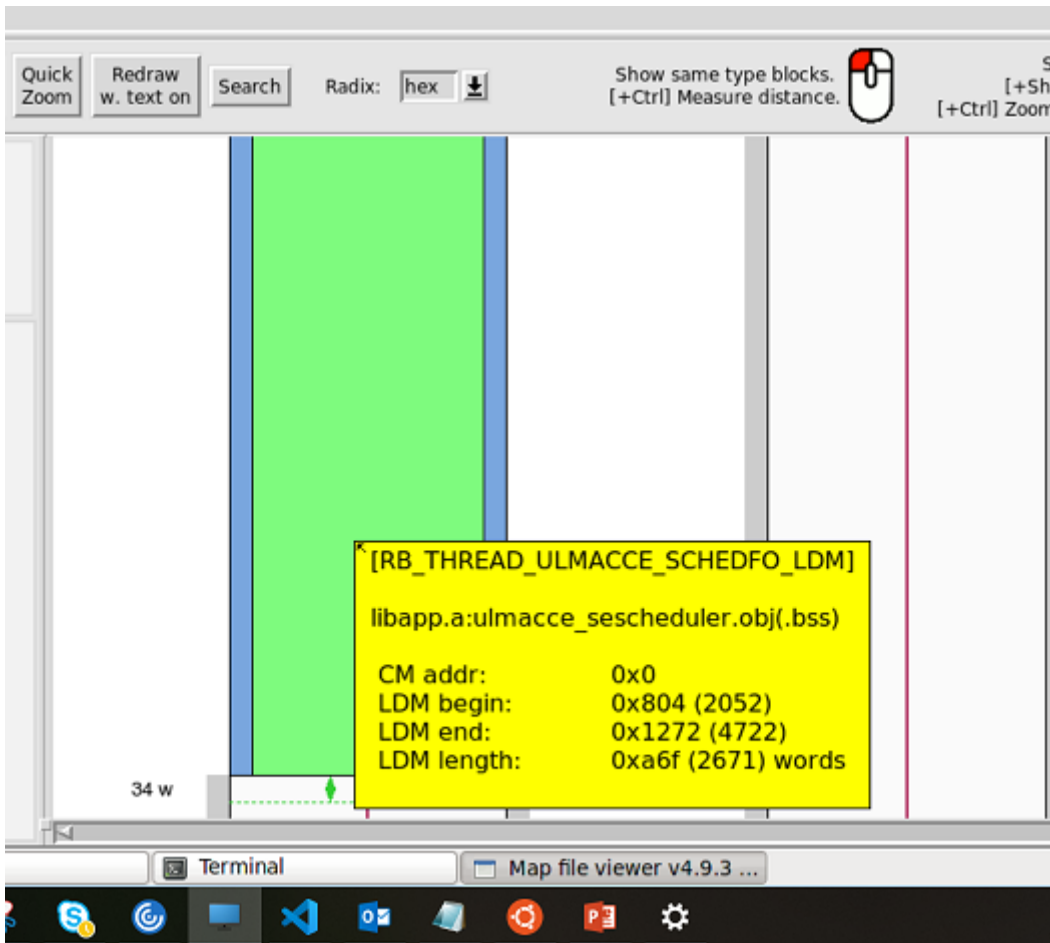
- error reported when building

```
ERROR: List of threads violating the heap & stack limit

Legend:
+ Requested Heap & stack violated.
* Heap & stack usage test margin violated.

Used heap Stack | Requested Heap Stack | Test Margin | Heap Stack limit | Thread Name
* 3352          | 3383                   | 34          | 3349              | test_ULMACCE_SCHEDFO
```

- memory map illustrated



# global variables refactor

*main approach is reducing its data type's size*

```
/* Add global variables that do NOT require initialisation within the following
 * # pragma */
#pragma nostdinit on
ULMACCE_SCHEDFO_foDataS ulSeSched_foData;
ULMACCE_PUSCHTPS_newTxSeDataS
ulSeSched_newSchedData[ELIB_BBBase_COMMON_MAX_NR_OF_UL_SE_PER_TT_PER_CELL];
UpcDlMacCeFiSePdcchCfmDataS
ulSeSched_pdcchCfmData[ELIB_BBBase_COMMON_MAX_NR_OF_UL_SE_PER_TT_PER_CELL];
ULMACCE_HARQPROCPOOLPPS_harqProcS
ulSeSched_harqProcData[ELIB_BBBase_COMMON_MAX_NR_OF_UL_SE_PER_TT_PER_CELL];
ULMACCE_SCHEDFO_internalSeDataS
    ulSeSched_internalSeData[ELIB_BBBase_COMMON_MAX_NR_OF_UL_SE_PER_TT_PER_CELL];
#pragma nostdinit off
```

## packing

```
typedef struct ULMACCE_NEW_newTxSeDataS
{
    U16 maxNrOfSbsForNew;  => 7bits
    S16 averageIslandNrOfSbs;=>7bits
}
```

## cutting

```
typedef struct ULMACCE_SCHEDFO_foDataS
{
    ULMACCE_admCtrlDataHeaderS admCtrlData;=>delte
}
when needed:
    ULMACCE_admCtrlDataHeaderS admCtrlData;
    ULMACCE_SHD_PUSCHTPS_getAdmCtrlDataHeader(&foData_p->puschTpsTo, &admCtrlData);
```

## if outside of struct

change flag(U16) int to bit, then using ULMACCE\_U16\_GET\_BIT\_VALUE()

## Pros & cons

- direct, easy to understand
- performance penalty
- few left to optimize

e.g

csiRequestBits actually need 3 bits only

below U16 var actually , but the change needs lots efforts, especially in test code which use it as U16.

## limitation of the compiler Flacc

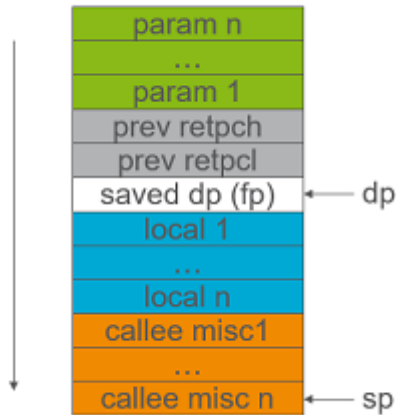
- can't pack signed data into U16/U32
- don't support anonymous union
- don't support c99 fully
- maybe new compiler clang change the situation

# local variables refactor

*stack resue is an edge tool*

## Flex Stack Frame Layout

- › **by caller:**
- › in parameters
- › saved retpc value (by call instruction)
- › **by callee:**
- › saved display pointer
- › local variables (static)
- › callee misc area (dynamic)



## a simple example in outline

```
void SESCHEDFO_main()//500+ loc
{
    U16 a;
    {
        U16 b;
    }

    {
        U16 c;
    }
    while(expr)
    {
        U16 d;
    }
}
```

## another example

*original code*

```

void SESCHEDFO_main()//500+ loc
{
    ULMACCE_LAPCSPLIB_traceS trace;
    ULMACCE_SCHEDLIB_traceHeaderS* traceHeader_p;

    traceHeader_p = &trace.header;
    traceHeader_p->cellId = foData_p->commonCellTo.cellInfo.cellId;

    while(expr)
    {
        trace.specificInfo.traceEntryInfo.nrofSchedulableSes = nrofSchedulableSes;

        BOOL isNs05seTypeDowngraded = FALSE;
        assignSpectrumSucceeded =
            ULMACCE_SCHEDLIB_assignSpectrumToSe(
                &trace
            );

        if (assignSpectrumSucceeded)
        {
            ULMACCE_PUSCHTPS_ginrWeighterDataS ginrWeightDataS;

        }
    }
}

```

***after refactor***

```

void SESCHEDFO_main()//500+ loc
{

while(expr)
{
    BOOL isNs05seTypeDowngraded = FALSE;
    {
        ULMACCE_LAPCSPLIB_traceS trace;
        ULMACCE_SCHEDLIB_traceHeaders* traceHeader_p;
        traceHeader_p = &trace.header;
        traceHeader_p->cellId = foData_p->commonCellTo.cellInfo.cellId;
        trace.specificInfo.traceEntryInfo.nrOfSchedulableSes = nrOfSchedulableSes;

        assignSpectrumSucceeded =
            ULMACCE_SCHEDLIB_assignSpectrumToSe(
                &trace
            );
    }

    if (assignSpectrumSucceeded)
    {
        ULMACCE_PUSCHTPS_ginrWeighterDataS ginrWeightDataS;

    }
}
}

```

result:

Decreased stack use in test\_ULMACCE\_SCHEDFO: by 26 words or 1.0%; new stack 2668 vs stack 2694

### ***assume***

***how much saved = how much resued - how much compiler alredy optimized***

## **Pros & cons**

- indirect & difficult to calculate
- more effort to read code
- no tool to find out the deepest call sequence in stack



# design refactor

e.g. any duplicate?

```
typedef struct ULMACCE_SESCHEDFO_seListS
{
    S16 size;
    S16 nrOfActiveElements;
    S16 startIndex;
    S16 iteratorIndex;
    ULMACCE_SESCHED_seDataS* se_p[ULMACCE_PUSCH_MAX_SE_LIST_SIZE];
} ULMACCE_SESCHEDFO_seListS;
```

```
typedef struct ULMACCE_SCHEDFO_foDataS
{
    ULMACCE_SCHEDFO_fodS fod;
    ULMACCE_SESCHEDFO_seListS inputNewSeList;
    ULMACCE_SESCHEDFO_seListS scheduledNewSeList;
} ULMACCE_SCHEDFO_foDataS;
```

in the fod, there is two list: inputNewSeList, scheduledNewSeList.

actually one se can only appear in 1 list, why need 2 space?

e.g. duplicate index?

```
#pragma nostdinit on
ULMACCE_SCHEDFO_foDataS ulSeSched_foData;
ULMACCE_PUSCHTPS_newTxSeDataS
ulSeSched_newSchedData[ELIB_BBBASE_COMMON_MAX_NR_OF_UL_SE_PER_TT_PER_CELL];
UpcdlMacCeFiSePdcchCfmDataS
ulSeSched_pdcchCfmData[ELIB_BBBASE_COMMON_MAX_NR_OF_UL_SE_PER_TT_PER_CELL];
ULMACCE_HARQPROCPOOLPPS_harqProcS
ulSeSched_harqProcData[ELIB_BBBASE_COMMON_MAX_NR_OF_UL_SE_PER_TT_PER_CELL];
ULMACCE_SCHEDFO_internalSeDataS
    ulSeSched_internalSeData[ELIB_BBBASE_COMMON_MAX_NR_OF_UL_SE_PER_TT_PER_CELL];
#pragma nostdinit off

{//loop
    newSchedData_p = ulSeSched_newSchedData;
    pdcchCfmData_p = ulSeSched_pdcchCfmData;
    harqProcData_p = ulSeSched_harqProcData;
    internalSeData_p = ulSeSched_internalSeData;
    newSchedData_p++;
    //and the other 3 pointer ++
}
```

merging 4 pointer into 1 index

```
static U16 seIndex = 0;

{ //loop
    newSchedData_p = &testdata_newSchedData[seIndex];
    harqProcData_p = &testdata_harqProcData[seIndex];
    pdcchCfmData_p = &testdata_pdcchCfmData[seIndex];
    internalSeData_p = &testdata_internalSeData[seIndex];
    seIndex++;
}
```