C Programming in z/OS
REGION and MEMLIMIT

In z/OS, the REGION and MEMLIMIT settings are used to limit the amount of virtual storage that is available for allocation.

- **REGION** refers to the user region and extended user region (i.e. below the 2-GB bar).
- **MEMLIMIT** refers to the virtual storage above the 2-GB bar.

The REGION parameter on the JOB or EXEC statement specifies an amount of storage in kilobytes (1 KB = 1024 bytes) or in megabytes (1 MB = 1024 KB):

```
>>> REGION={valueK|valueM} <<<
```

- The maximum value you can specify is **2096128K** or **2047M**.
- A value equal to **0K** or **0M** gives all the storage available below and above the 16-MB line (but below the 2-GB bar).
  - When **0K** or **0M** is specified for REGION, MEMLIMIT is set to NOLIMIT.
- A non-zero value less than or equal to **16384K** (or **16M**) establishes the size of the region below the 16-MB line. The size of the region above the 16-MB line (and below the 2-GB bar) is given the default value of 32 MB (32768 KB).
- A value greater than **16384K** (or **16M**) and less than or equal to **32768K** (or **32M**) gives all the storage available below the 16-MB line. The size of the region above the 16-MB line (and below the 2-GB bar) is given the default value of 32 MB (32768 KB).
- A value greater than **32768K** (or **32M**) establishes the size of the region above the 16-MB line (and below the 2-GB bar), and gives all the storage available below the 16-MB line.

---

1 These amounts are installation dependent.
2 See further.
Note that your installation can use the **IEFUSI** or **IEALIMIT** exit\(^1\) to override the value specified on the **REGION** parameter.

If no **REGION** parameter is specified, the system uses a jobclass dependent default\(^2\).

The **MEMLIMIT** parameter on the **JOB** or **EXEC** statement specifies:

- either an amount of storage in megabytes (1 MB = 1024 KB), gigabytes (1 GB = 1024 MB), terabytes (1 TB = 1024 GB), or petabytes (1 PB = 1024 TB); this value may be from 0 to 99999, with a maximum of 16384P
- or **NOLIMIT**

\[\text{MEMLIMIT=\{valueM|valueG|valueT|valueP|NOLIMIT\}}\]

- Unlike the **REGION** parameter, **MEMLIMIT=0M** (or equivalent in G, T, or P) means that you cannot allocate virtual storage above the 2-GB bar.
- If no **MEMLIMIT** parameter is specified, the default is the value defined to SMF\(^3\), except when **REGION=0K** (or equivalent in M) is specified, in which case the default is **NOLIMIT**.
  - You can use the z/OS operator command **DISPLAY SMF,0** to display the **MEMLIMIT** value assigned by SMF.
  - Note that your installation can use the **IEFUSI** exit to override any JCL- or SMF-supplied **MEMLIMIT** value\(^4\).
  - The **MEMLIMIT** for z/OS UNIX users can be specified via the RACF OMVS segment\(^5\).
  - **MEMLIMIT=NOLIMIT** is equivalent to **MEMLIMIT=16384P** (which is the maximum value).
    - For **MEMLIMIT=NOLIMIT** and **MEMLIMIT=16384P**, the contents of the 64-bit **RAXLVMEMLIM** field\(^6\) is 00000FFFFFFFF00016 (which represents 17592186044416 MB\(^7\)).

Note that the TSO/E **LOGON** screen provides no field to enter a **MEMLIMIT** value.

- You can use the **Size** field to specify the **REGION** size.
- Once logged on, you can change the **MEMLIMIT** value via z/OS UNIX\(^6\).

Beginning with z/OS V1R2, SDSF provides the **MemLimit** column on the **Display Active Users (DA)** panel.

---

\(^1\) See the “Installation Exits” book in the z/OS MVS bookshelf.

\(^2\) See the “Installation and Tuning Reference” in the z/OS JES2 bookshelf.

\(^3\) See the “System Management Facilities” book in the z/OS MVS bookshelf.


\(^5\) See the “UNIX System Services Planning” book in the z/OS UNIX bookshelf.

\(^6\) See further.

\(^7\) 16384 PB is in fact equivalent to 17592186044416 MB.
## Exercise 9

The “Data Areas” volumes, which reside in the z/OS MVS bookshelf, describe the many data areas (control blocks) used by z/OS and by application programs.

For this exercise, we will use the following areas:

<table>
<thead>
<tr>
<th>address</th>
<th>field name</th>
<th>contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0_{16}$</td>
<td>Prefixed Save Area</td>
<td>addresses of various control blocks</td>
</tr>
<tr>
<td>PSA+$22_{16}$</td>
<td>PSAAOLD</td>
<td>4-byte pointer to the current Address Space Control Block (ASCB)</td>
</tr>
<tr>
<td>ASCB+$24_{16}$</td>
<td>ASCBASID</td>
<td>2-byte signed integer specifying the Address Space Identifier (ASID)</td>
</tr>
<tr>
<td>ASCB+$30_{16}$</td>
<td>ASCBLDA</td>
<td>4-byte pointer to the Local Data Area (LDA)</td>
</tr>
<tr>
<td>LDA+$E8_{16}$</td>
<td>LDALOAL</td>
<td>4-byte signed integer specifying the amount of REGION storage (in bytes) currently in use (allocated) below the 16-MB line</td>
</tr>
<tr>
<td>LDA+$D0_{16}$</td>
<td>LDALIMIT</td>
<td>4-byte signed integer specifying the REGION limit value (in bytes) for virtual storage below the 16-MB line</td>
</tr>
<tr>
<td>LDA+$F0_{16}$</td>
<td>LDAELOAL</td>
<td>4-byte signed integer specifying the amount of REGION storage (in bytes) currently in use (allocated) above the 16-MB line (and below the 2-GB bar)</td>
</tr>
<tr>
<td>LDA+$D8_{16}$</td>
<td>LDAEELIM</td>
<td>4-byte signed integer specifying the REGION limit value (in bytes) for virtual storage above the 16-MB line (and below the 2-GB bar)</td>
</tr>
<tr>
<td>ASCB+$16C_{16}$</td>
<td>ASCBRSME</td>
<td>4-byte pointer to the Real Storage Manager Address Space Block Extension (RAX)</td>
</tr>
<tr>
<td>RAX+$98_{16}$</td>
<td>RAXLVMEMLIMIT</td>
<td>8-byte signed integer specifying the MEMLIMIT value (in MB) for virtual storage above the 2-GB bar</td>
</tr>
</tbody>
</table>

Write, compile, and execute a z/OS XL C AMODE 64 program to print ASCBASID, LDALOAL, LDALIMIT, LDAELOAL, LDAELIM, and RAXLVMEMLIMIT.
Exercise 9 - solution

Our C program source:

```c
#include <stdio.h>

int main() /* C99 allows mixing declarations with executable code */
{
    int address_of_ASCB = * (int *) 0x224;
    int address_of_ASCBASID = address_of_ASCB + 0x24;
    short int ASCBASID = * (short int *) address_of_ASCBASID;
    printf("This address space's ASID is %hd (hexadecimal %02hX)\n",
           ASCBASID, ASCBASID);
    int address_of_LDA = * (int *) (address_of_ASCB + 0x30);
    int address_of_LDAELOAL = address_of_LDA + 0xE8;
    int LDAELOAL = * (int *) address_of_LDAELOAL;
    printf("Region BELOW the 16-MB line:\n"
           "- currently in-use: %13d KB\n", LDAELOAL / 1024);
    int address_of_LDAELIM = address_of_LDA + 0xD8;
    int LDAELIM = * (int *) address_of_LDAELIM;
    printf("- limit value: %13d KB\n", LDAELIM / 1024);
    printf("- free (available): %13d KB\n", LDAELIM / 1024 - LDAELOAL / 1024);
    int address_of_RAX = * (int *) (address_of_ASCB + 0x16C);
    int address_of_RAXLVMEMLIM = address_of_RAX + 0x98;
    long long int RAXLVMEMLIM = *
         *(long long int *) address_of_RAXLVMEMLIM;
    printf("Large memory ABOVE the 2-G bar:\n"
           "- MEMLIMIT value: %13lld MB\n", RAXLVMEMLIM);
    return 0;
}
```

To compile, bind, and execute the AMODE 64 program in z/OS batch:

```plaintext
//jobname JOB 'M. CASTELEIN',CLASS=A,MSGCLASS=X, z/OS V1R8
// COND=(0,NE),NOTIFY=&SYSUID
//*-----------------------------------------------------------------
//COMPILE EXEC PGM=CCNDRVR,REGION=96M, z/OS XL C V1.8
// PARM='SOURCE,LP64'
//STEPLIB DD DSN=CEE.SCEERUN2,DISP=SHR
// DD DSN=CBC.SCCNCMP,DISP=SHR
// DD DSN=CEE.SCEERUN,DISP=SHR
```

©Copyright Arcis Services 2005-2009
The **SYSPRINT** output for step **GO** is:

```
This address space's ASID is 24 (hexadecimal 18)
Region BELOW the 16-MB line:
- currently in-use: 12 KB
- limit value: 9192 KB
```
The preceding JCL is based on the IBM-supplied procedure EDCQCBG.

In z/OS batch, the IBM-supplied default for z/OS XL C is `LANGLVL(EXTENDED)` which enables IBM extensions\(^1\) to the C89 standard.

- You must use `LANGLVL(STD C99)` or `LANGLVL(EXT C99)` if your C program source contains the `inline` keyword, `restrict` keyword, and/or C++ comments.

- If the LP64 compiler option is in effect, z/OS XL C uses the XPLINK compiler option by default.
  - An XPLINK-compiled program is implicitly a DLL-compiled program.

- If LP64 or XPLINK is in use, z/OS XL C uses the the GOFF compiler option by default.

- Recall that every z/OS XL C AMODE 64 program must be a program object\(^2\), i.e. it must be stored in a PDSE program library or z/OS UNIX file.

The execution of an AMODE 64 program under TSO/E may fail because of the MEMLIMIT setting\(^3\):

---

**Note:**

\(^1\) See further.

\(^2\) Load modules are stored in PDS program libraries.

\(^3\) You can change the MEMLIMIT value for a TSO/E address space via z/OS UNIX. See further.
MEMLIMIT in z/OS UNIX and TSO/E

If a z/OS program does not access storage above the 2-GB bar, there is no need to run it in AMODE 64:

- In z/OS UNIX, every shell session runs in a separate BPXAS address space which is provided by the Workload Manager (WLM).
- When a shell session is created using the TSO/E OMVS command, the BPXAS address space inherits the REGION and MEMLIMIT settings from the parent (i.e. the user's) TSO/E address space.
  - The RACF security administrator may override this MEMLIMIT setting by defining a MEMLIMIT value for individual users in the associated RACF OMVS segment.
- To change the MEMLIMIT of a specific UNIX process (or to be more exact, to change the MEMLIMIT value of the address space that the process is running in), you can use the z/OS operator command SETOMVS:
  1. Enter the `ps` shell command to display status information about the processes associated with your terminal.
  2. You can use the z/OS operator command DISPLAY OMVS to display information about the current limits for a process with a given process identifier (PID).
3. Use the z/OS operator command **SETOMVS** to change the **MEMLIMIT** value of the **BPXAS** address space.
4. You can reenter the z/OS operator command **DISPLAY OMVS** to check that the limit has changed.

In response to the **DISPLAY OMVS** operator command, the system displays:

```plaintext
DISPLAY OMVS, LIMITS, PID=50397199 /* specify the PID for /bin/sh */
BPX0051I 13.30.31 DISPLAY OMVS 594
OMVS 000E ACTIVE OMVS=(CS)
USER JOBNAME ASID PID PPID STATE START CT SECS
IBMUSER IBMUSER 0034 50397199 33619974 1C---- 13.09.29 5.926
LATCHWAITPID= 0 CMD=sh -L
PROCESS LIMITS: LIMMSG=NONE
CURRENT USAGE CURRENT USAGE LIMIT
MAXFILEPROC 6 6 400
MAXFILESIZE --- --- NOLIMIT
MAXPROCUSER 18 22 NOLIMIT
MAXQUEUESIGS 0 1 1000
MAXTHREADS 0 0 10000
MAXTHREADTASKS 0 0 5000
IPCSHMNSEGS 0 0 500
MAXCORESIZE --- --- 4194304
MAXMEMLIMIT 0 0 0
```

**In response to the DISPLAY OMVS operator command,** the system displays:

- **Currently in-use**: 64 KB
- **Limit value**: 9192 KB
- **Free (available)**: 9128 KB
- **Region above the 16-MB line**: 344 KB
- **Limit value**: 1806336 KB
- **Free (available)**: 1805992 KB
- **Large memory above the 2-G bar**: 0 MB

**IBMUSER:/usr/ccar >c99 -Wc,LPE64 -o storage64 storage.c**

**IBMUSER:/usr/ccar >file storage64**

**storage64**: z/OS Unix executable (amode=64)

**IBMUSER:/usr/ccar >storage64**

`'T' + Done(137) storage64` 88951649  Killed ./storage64

**IBMUSER:/usr/ccar >ps**

`50397199 ttup0000 0:04 /bin/sh 88951650 ttup0000 0:00 /bin/ps` 29397199

**IBMUSER:/usr/ccar >**

```plaintext
TID TTY TIME CMD
50397199 ttup0000 0:04 /bin/sh
88951650 ttup0000 0:00 /bin/ps
```

**use SDSF option LOG to check the system log**: message IEF450I shows **ABEND** completion code U4093 REASON=00000224
Change the **MEMLIMIT** value of the address space:

<table>
<thead>
<tr>
<th>Command</th>
<th>PID</th>
<th>MEMLIMIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETOMVS PID=50397199,MEMLIMIT=20M</td>
<td>50397199</td>
<td>20M</td>
</tr>
</tbody>
</table>

BPX0015I THE SETOMVS COMMAND WAS SUCCESSFUL.

BPX0051I 13.38.05 DISPLAY OMVS 598

```
OMVS     000E ACTIVE
USER     JOBNAME  ASID        PID       PPID STATE   START     CT_SECS
IBMUSER  IBMUSER  0034   50397199   33619974 1C---- 13.09.29     6.738

LATCHWAITPID=         0 CMD=sh -L
PROCESS LIMITS:       LIMMSG=NONE
                       CURRENT        HIGHWATER        PROCESS
                       USAGE        USAGE        LIMIT
MAXFILEPROC            6       6             400
MAXFILESIZE            ---      ---             NOLIMIT
MAXPROCUSER            18      22             NOLIMIT
MAXQUEUESIGS           0       1             1000
MAXTHREADS             0       0             10000
MAXTHREADTASKS         0       0             500
IPCSHMNSEGS             0       0             500
MAXCORESIZE            ---      ---             4194304
MAXMEMLIMIT             0       0          20M *
```

- An * at the end of a row indicates that this value has been changed by a SETOMVS (or SET OMVS) operator command.

This address space's ASID is 49 (hexadecimal 31)

Region BELOW the 16-MB line:
- **Currently in-use:** 4 KB
- **Limit value:** 9192 KB
- **Free (available):** 9188 KB

Region ABOVE the 16-MB line:
- **Currently in-use:** 6672 KB
- **Limit value:** 1800336 KB
- **Free (available):** 1793664 KB

Large memory ABOVE the 2-G bar:
- **MEMLIMIT value:** 20 MB

--->

The program execution is now successful.
Note that the new **MEMLIMIT** value for the z/OS UNIX shell session also applies to the TSO/E session that initiated the shell session.

- Review the preceding **BPXO051I** messages: **ASID 003416** refers to the user's TSO/E address space.

Alternatively, you can use the `ps` and `ulimit` shell commands to display **MEMLIMIT**:

- In z/OS UNIX, you can specify the `-o` option on the `ps` command to define the status fields that will be displayed: **pid** displays the process ID, **vsz** displays the amount of virtual storage (in kilobytes) the process is using below the 2-GB bar, **vsz64** displays the amount of virtual storage used above the 2-GB bar, **vszlmt64** displays the **MEMLIMIT** value, **xasid** displays the ASID as a hexadecimal value, etc.

- Beginning with **z/OS V1R8**, you can in z/OS UNIX use the `-M` option on the `ulimit` command to display the **MEMLIMIT** value (in megabytes).

---

1. **Parent PID (PPID)** 33619974 refers to the TSO/E OMVS command.
2. **vsz64** and **vszlmt64** are available beginning with **z/OS V1R6**.
3. See the “Command Reference” in the **z/OS UNIX System Services** bookshelf.