**eXpOS Report**

**Stage 2:**

**Question 1:**

When a file is created entries are made in the Inode table as well as the Root file. What is the need for this duplication

**Solution:**

Inode table is a data structure which is accessible only in Kernel mode, whereas Root file is accessible both in Kernel and User mode. This enables the user to search for a file from an application program itself by reading the Root file.

**Assignment 1:**

Copy the contents of Root File (from Block 5 of XFS disk) to a UNIX

file $HOME/myexpos/root\_file.txt and verify that an entry for sample.dat is made in it also.

**Solution:**

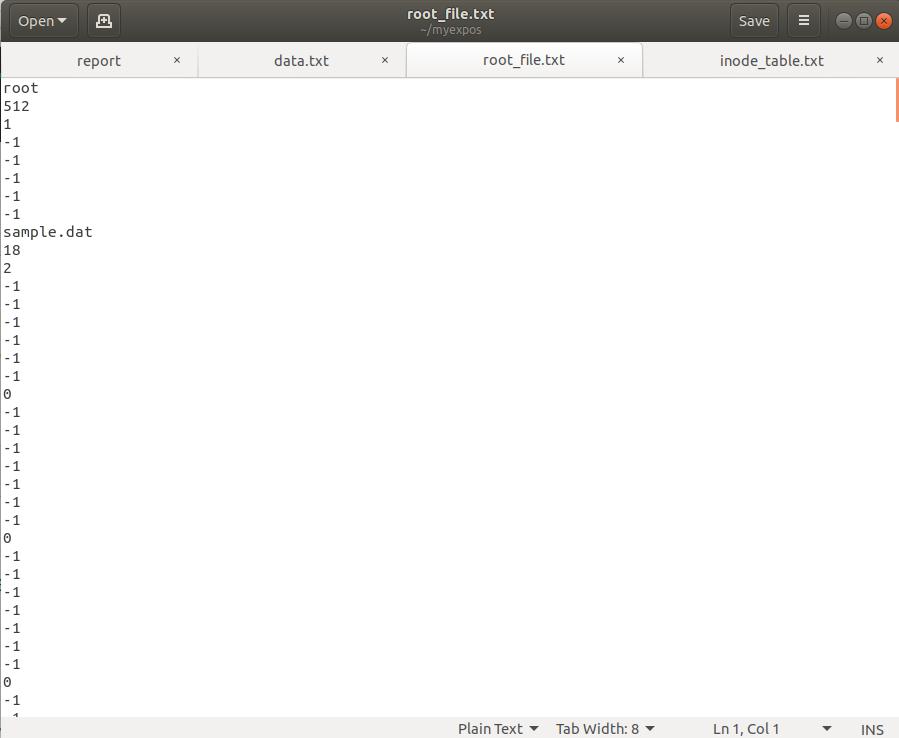
First we need to open the xfs interface using the following command in the terminal with home directory being

./xfs-interface

Using the Copy command in xfs interface as follows

copy 5 5 $HOME/myexpos/root\_file.txt

Now the data in root\_file.txt is changed as follows:



**Assignment 2:**

Delete the sample.dat from the XSM machine using xfs-interface and note the changes for the entries for this file in *inode table, root file and disk free list*.

**Solution:**

First let’s remove the sample.dat using the following command

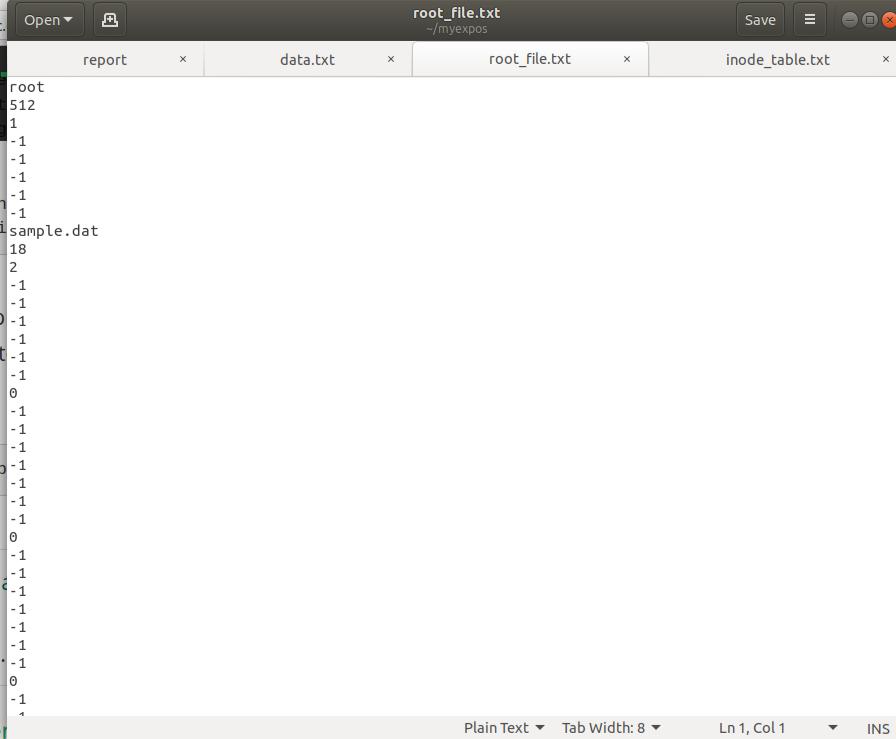
rm sample.dat

To see the changes in root file and inode table, we need to use these commands in the xfs-interface:

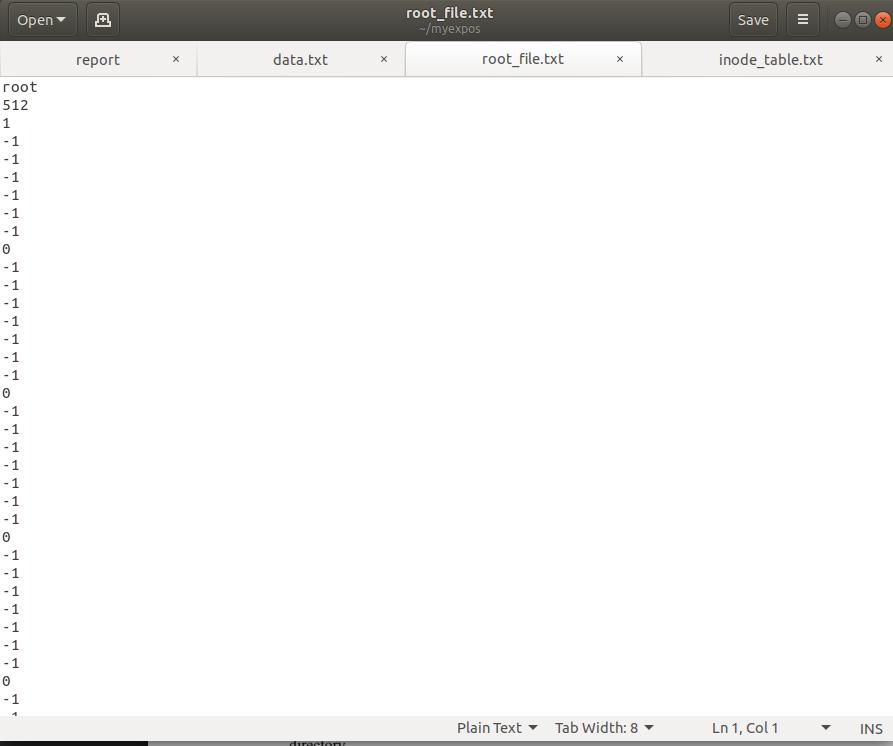
copy 3 4 $HOME/myexpos/inode\_table.txt copy 5 5 $HOME/myexpos/root\_file.txt

Changes in root\_file.txt are as follows.

Before:

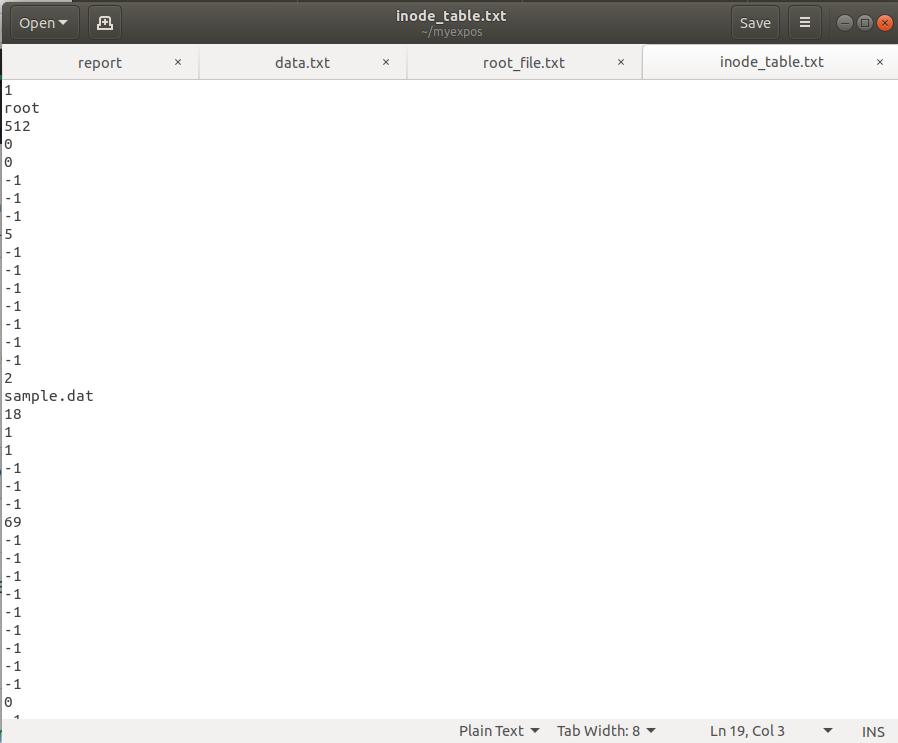


After:

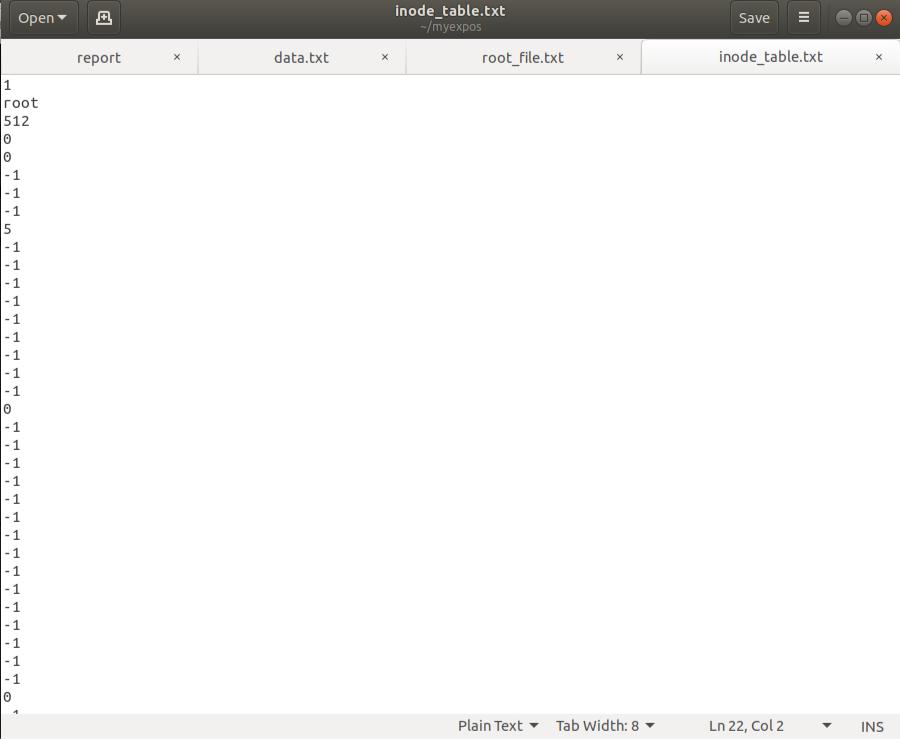


Changes in inode\_table.txt are as follows

Before:



After:



Changes in disk free list are as follows:

Before:



After:



We can observe that the 69th block changed to ‘0’.

**Stage 3:**

**Question 1:**

**Q1.** If the OS Startup Code is loaded to some other page other than Page 1, will XSM work fine?

**Solution:**

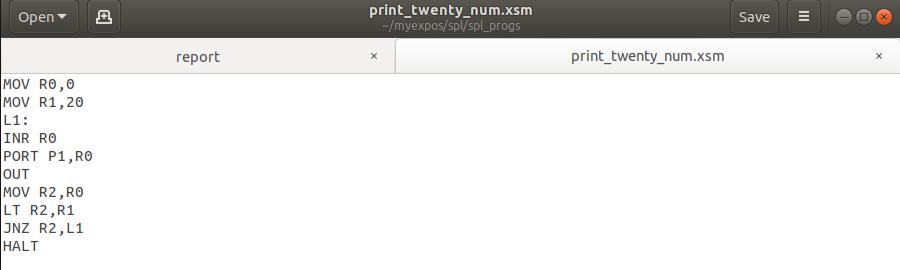
No. This is because after the execution of the ROM Code, IP points to **512** which is the 1st instruction of Page 1. So, if the OS Startup Code is not loaded to Page 1, it results in an exception and leads to system crash.

**Assignment 1:**

Write an assembly program to print numbers from 1 to 20 and run it as the OS Startup code.

**Solution:**

Write and Save the program print\_twenty\_num.xsm as follows in spl/spl\_progs:



Then use the following command in xfs interface as follows:

* load --os $HOME/myexpos/spl/spl\_progs/print\_twenty\_num.xsm
* exit

Now for the program to be executed go to xsm folder in terminal and use the following command

./xsm

To get the following results



**Stage 4:**

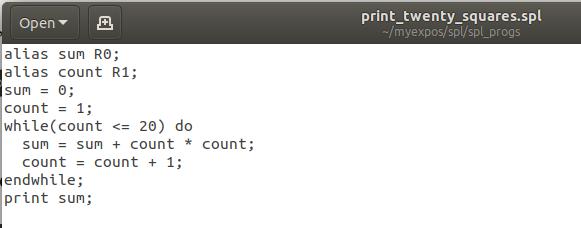
**Assignment 1:**

Write the spl program to print sum of squares of the first 20 natural numbers.

Load it using xfs interface and run the in the machine.

**Solution:**

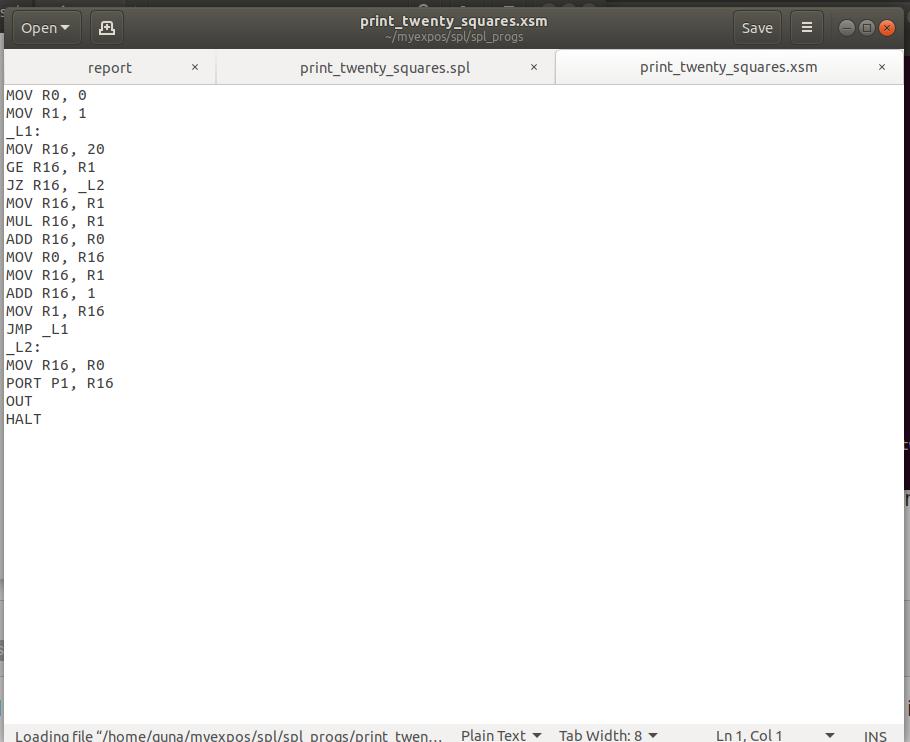
Write and Save the program print\_twenty\_squares.spl as follows in spl/spl\_progs:



Compile the following program with spl compiler using the following command

./spl spl\_progs/print\_twenty\_squares.spl

This has generated a new file in spl\_progs as print\_twenty\_squares.xsm as:



Now load the load the program in xfs interface like we did before and exit and change directory to xsm and run ./xsm to get the following result.



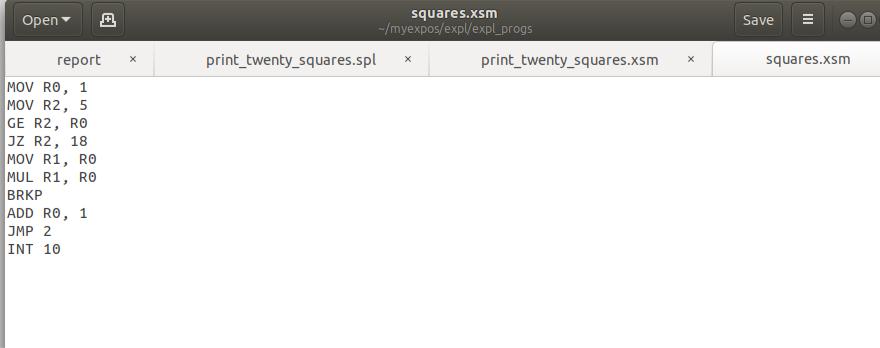
**Stage 6:**

**Assignment 1:**

Change virtual memory model such that code occupies logical pages 4 and 5 and the stack lies in logical page 8. You will have to modify the user program as well as the os startup code.

**Solution:**

User Code:



Save this code in eps/eps\_progs as squares.xsm and load it in xfs interface as follows

load --init $HOME/myexpos/expl/expl\_progs/squares.xsm

Create another file for halt with just one line “halt;” and save it as halt.spl in spl\_progs and then compile it with spl compiler and load it as follows

load --int=10 $HOME/myexpos/spl/spl\_progs/halt.xsm load --exhandler $HOME/myexpos/spl/spl\_progs/halt.xsm

Create an os startup file as follows



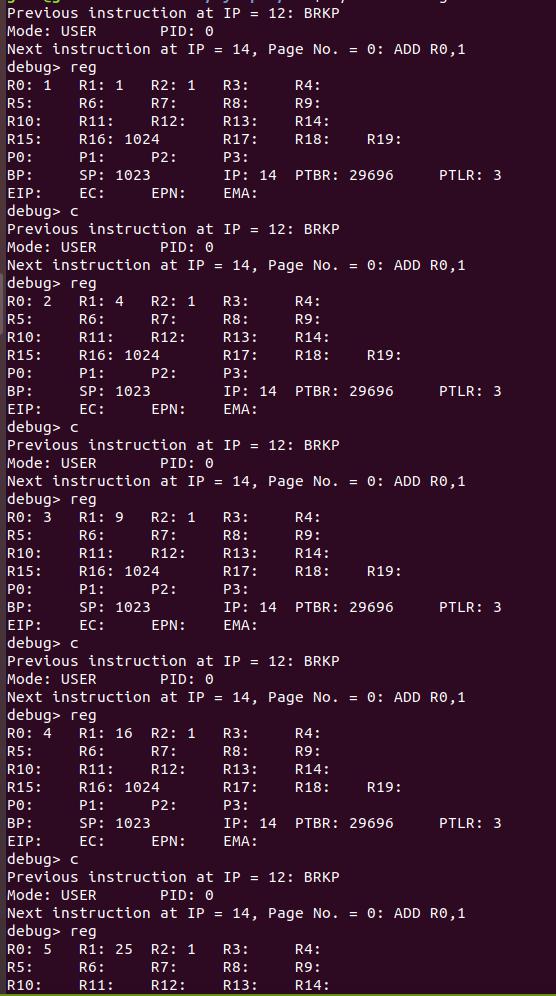
Save it as os\_startup.spl and then compile it spl compiler as discussed before and load in it xfs as follows

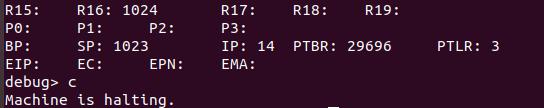
load --os $HOME/myexpos/spl/spl\_progs/osstartup.xsm

Exit the interface and then use the follow command to run the program

./xsm --debug --timer 0

At each step of entering reg observe the value in register R1 at each breakpoint





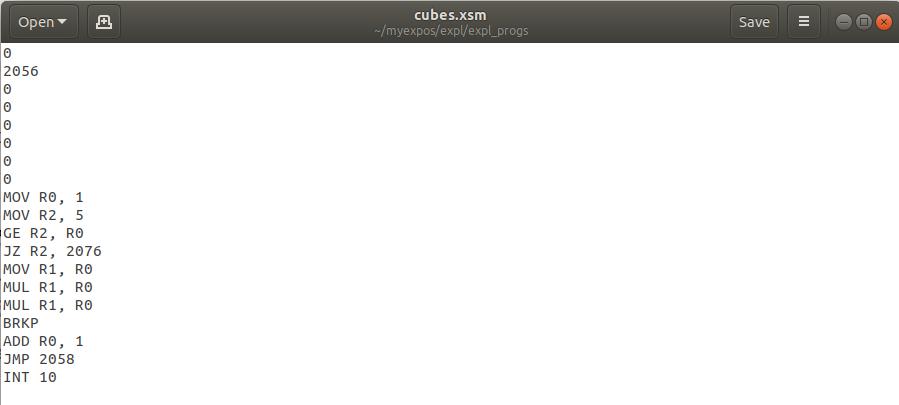
**Stage 7:**

**Assignment 1:**

Change the user program to compute cubes of the first five numbers.

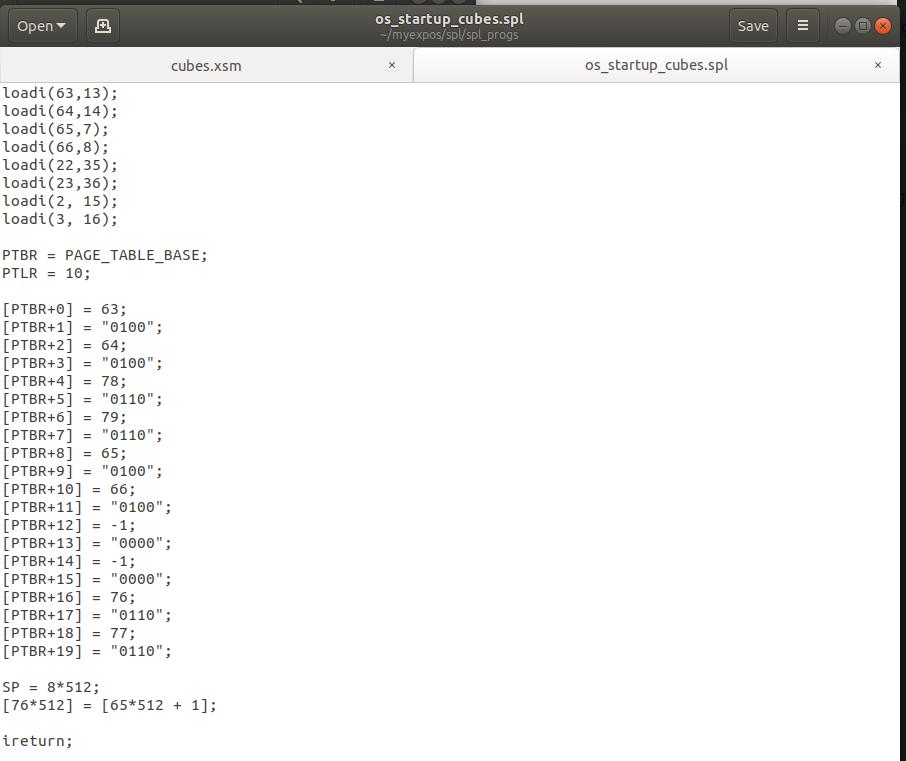
**Solution:**

Create a new file with name cubes.xsm and type the following code in it



Save this code in eps/eps\_progs as cube.xsm and load it in xfs interface as follows load --init $HOME/myexpos/expl/expl\_progs/cubes.xsm

Now let’s create an another os startup file with name os\_startup\_cubes.spl and save it in spl\_progs.



Compile it with spl compiler just like in stage 6 and load it in xfs interface and exit and run the program by following command

./xsm --debug --timer 0

At each step of entering reg observe the value in register R1 at each breakpoint

