RARS Environment calls

RARS currently supports system calls that MARS originally supported and system calls compatible with Linux tooling (gcc, riscv-pk, etc). Only a handful are compatible with other education simulators, but it is configurable so the call numbers can be changed with a simple config file.

They can be called by loading the call number into a7, any other arguments into a0-a6 and calling ecall. The following exits the program with the code 42.

```
li a0, 42
li a7, 93  # code to exit the program: 93
ecall
```

```
     .data
str: .string "abc"

    .text
    li a0, 5
    li a7, 1   # code to print an integer: 1
    ecall    # prints "5" to console

    li a0, 42
    li a7, 11   # code to print a char : 11
    ecall    # prints "*" to console (ASCII(42): '*')

    la a0,  str  # load address of "str" in a0
    li a7, 4    #  # code to print a string: 4
    ecall     # prints "abc" to console
```

The register a7 contains the system call number (ecall). The register a0 contains the parameter from/to the ecall. All supported system calls, with respective parameters are shown below.

Name	Call # (a7)	Description	Inputs	Outputs
PrintInt	1	Prints an integer	a0: integer to print	none
PrintFloat	2	Prints an floating point number	fa0: float to print	none
PrintString	4	Prints a null-terminated string to the console	a0: the address of the string	none
ReadInt	5	Read an int from input console	none	a0: the int
ReadFloat	6	Read a float from input console	none	fa0: the float

Name	Code	Description	Inputs	Outputs
ReadString	8	Reads a string from the console	a0: address of input buffer a1: maximum number of characters to read	none
Sbrk	9	Allocate heap memory	a0: amount of memory in bytes	a0: address to the allocated block
Exit	10	Exits the program with code 0	none	none
PrintChar	11	Prints an ascii character	a0: character to print (only lowest byte is considered)	none
ReadChar	12	Read a character from input console	none	a0: the character
GetCWD	17	Writes the path of the current working directgory into a buffer	a0: the buffer to write into a1: the length of the buffer	a0: -1 if the path is longer than the buffer
Time	30	Get the current time (milliseconds since 1 January 1970)	none	a0: low order 32 bits high order 32 bits
MidiOut	31	Outputs simulated MIDI tone to sound card (does not wait for sound to end).	See MIDI note below	none
Sleep	32	Set the current thread to sleep for a time (not precise)	a0: time to sleep in milliseconds	none
MidiOutSync	33	Outputs simulated MIDI tone to sound card, then waits until the sound finishes playing.	See MIDI note below	none
PrintIntHex	34	Prints an integer (in hexadecimal format left-padded with zeroes)	a0: integer to print	none
PrintIntBinary	35	Prints an integer (in binary format left-padded with zeroes)	a0: integer to print	none
PrintIntUnsigned	36	Prints an integer (unsigned)	a0: integer to print	none
RandSeed	40	Set seed for the underlying Java pseudorandom number generator	a0: index of pseudorandom number generator a1: the seed	none
RandInt	41	Get a random integer	a0: index of pseudorandom number generator	a0: random integer
RandIntRange	42	Get a random bounded integer	a0: index of pseudorandom number generator a1: upper bound for random number	a0: uniformly selectet from [0,bound]
RandFloat	43	Get a random float	a0: index of pseudorandom number generator	fa0: uniformly randomly selected from from [0,1]
ConfirmDialog	50	Service to display a message to user	a0: address of null-terminated string that is the message to user	a0: Yes (0), No (1), or Cancel(2)

InputDialogInt	51	none	none	none
InputDialogFloat	52	none	none	none
InputDialogString	54	none	none	none
MessageDialog	55	Service to display a message to user	a0: address of null-terminated string that is the message to user a1: the type of the message to the user, which is one of: 1: error message 2: information message 3: warning message 4: question message other: plain message	none
MessageDialogInt	56	Service to display a message followed by a int to user	a0: address of null-terminated string that is the message to user a1: the int to display	none
Close	57	Close a file	a0: the file descriptor to close	none
MessageDialogFloat	58	Service to display a message followed by a float to user	a0: address of null-terminated string that is the message to user fa1: the float to display	none
MessageDialogString	59	Service to display a message followed by a string to user	a0: address of null-terminated string that is the message to user a1: address of the second string to display	none
LSeek	62	Seek to a position in a file	a0: the file descriptor a1: the offset for the base a2 is the begining of the file (0), the current position (1), or the end of the file (2)}	a0: the selected position from the beginning of the file or -1 is an error occurred
Read	63	Read from a file descriptor into a buffer	a0: the file descriptor a1: address of the buffer a2: maximum length to read	a0: the length read or -1 if error
Write	64	Write to a filedescriptor from a buffer	a0: the file descriptor a1: the buffer address a2: the length to write	a0: the number of charcters written
Exit2	93	Exits the program with a code	a0: the number to exit with (ignored in the gui)	none
Open	1024	Opens a file from a path Only supported flags (a1) are read-only (0), write-only (1) and write-append (9). write-only flag creates file if it does not exist, so	a0: Null terminated string for the path a1: flags	a0: the file descriptor or -1 if an error occurred

		it is technically write-create. write-append will start writing at end of existing file.		