Appendix A - System Calls
This chapter is based on the StarkNet documentation available at [StarkNet Docs](https://docs.starknet.io/documentation/architecture_and_concepts/Smart_Contracts/system-calls-cairo1/).
Writing smart contracts requires various associated operations, such as calling another contract or accessing the contract’s storage, that standalone programs do not require. The Starknet contract language supports these operations by using system calls. System calls enable a contract to require services from the Starknet OS. You can use system calls in a function to get information that depends on the broader state of Starknet, which would otherwise be inaccessible, rather than local variables that appear in the function’s scope.
Here is a list of the system calls available in Cairo 1.0:
- [get_block_hash](#get_block_hash)
- [get_execution_info](#get_execution_info)
- [call_contract](#call_contract)
- [deploy](#deploy)
- [emit_event](#emit_event)
- [library_call](#library_call)
- [send_message_to_L1](#send_message_to_l1)
- [replace_class](#replace_class)
- [storage_read](#storage_read)
- [storage_write](#storage_write)
- [sha256_process_block](#sha256_process_block)
`get_block_hash`
Syntax
```cairo
extern fn get_block_hash_syscall(
    block_number: u64
) -> SyscallResult<felt252> implicits(GasBuiltin, System) nopanic;
```
Description
Gets the hash of a specific StarkNet block within the range of `[first_v0_12_0_block, current_block - 10]`.
Return Values
Returns the hash of the given block.
Error Messages
- `Block number out of range`: `block_number` is greater than _`current_block`_`- 10`.
- `0`: `block_number` is less than the first block number of v0.12.0.
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/0c882679fdb24a818cad19f2c18decbf6ef66153/corelib/src/starknet/syscalls.cairo#L37)
`get_execution_info`
Syntax
```cairo
extern fn get_execution_info_syscall() ->
```

SyscallResult<Box<starknet::info::ExecutionInfo>> implicits(
    GasBuiltin, System
) nopanic;
~~~
#### Description
Gets information about the original transaction.
In Cairo 1.0, all block/transaction/execution context getters are batched into this single system call.
#### Arguments
None.
#### Return Values
Returns a [struct](https://github.com/starkware-libs/cairo/blob/efbf69d4e93a60faa6e1363fd0152b8fcedbb00a/corelib/src/starknet/info.cairo#L8) containing the execution info.
#### Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/cca08c898f0eb3e58797674f20994df0ba641983/corelib/src/starknet/syscalls.cairo#L35)
## `call_contract`
#### Syntax
```cairo
extern fn call_contract_syscall(
    address: ContractAddress, entry_point_selector: felt252, calldata: Span<felt252>
) -> SyscallResult<Span<felt252>> implicits(GasBuiltin, System) nopanic;
~~~
Description
Calls a given contract. This system call expects the address of the called contract, a selector for a function within that contract, and call arguments.
> **Note:**
>
> An internal call can’t return Err(_) as this is not handled by the sequencer and the Starknet OS.
>
> If call_contract_syscall fails, this can’t be caught and will therefore result in the entire transaction being reverted.
Arguments
- _`address`_: The address of the contract you want to call.
- _`entry_point_selector`_: A selector for a function within that contract, can be computed with the `selector!` macro.
- _`calldata`_: The calldata array.
Return Values
The call response, of type `SyscallResult<Span<felt252>>`.
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/cca08c898f0eb3e58797674f20994df0ba641983/corelib/src/starknet/syscalls.cairo#L10)
> **Note:**
> This is considered a lower-level syntax for calling contracts.

> If the interface of the called contract is available, then you can use a more straightforward syntax.
`deploy`
Syntax
```cairo
extern fn deploy_syscall(
    class_hash: ClassHash,
    contract_address_salt: felt252,
    calldata: Span<felt252>,
    deploy_from_zero: bool,
) -> SyscallResult<(ContractAddress, Span::<felt252>)> implicits(GasBuiltin, System) nopanic;
```
Description
Deploys a new instance of a previously declared class.
Arguments
- _`class_hash`_: The class hash of the contract to be deployed.
- _`contract_address_salt`_: The salt, an arbitrary value provided by the sender. It is used in the computation of the contract's address.
- _`calldata`_: The constructor's calldata. An array of felts.
- _`deploy_from_zero`_: A flag used for the contract address computation. If not set, the caller address will be used as the new contract's deployer address, otherwise 0 is used.
Return Values
A tuple wrapped with SyscallResult where:
- The first element is the address of the deployed contract, of type `ContractAddress`.
- The second element is the response array from the contract's constructor, of type `Span::<felt252>`.
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/4821865770ac9e57442aef6f0ce82edc7020a4d6/corelib/src/starknet/syscalls.cairo#L22)
`emit_event`
Syntax
```cairo
extern fn emit_event_syscall(
    keys: Span<felt252>, data: Span<felt252>
) -> SyscallResult<()> implicits(GasBuiltin, System) nopanic;
```
Description
Emits an event with a given set of keys and data.
For more information and a higher-level syntax for emitting events, see [Starknet events](https://docs.starknet.io/documentation/architecture_and_concepts/Smart_Contracts/starknet-events/).
Arguments
- _`keys`_: The event's keys. These are analogous to Ethereum's event topics, you can use the starknet_getEvents method to filter by these keys.

- _`data`_: The event’s data.
Return Values
None.
Example
The following example emits an event with two keys, the strings `status` and `deposit` and three data elements: `1`, `2`, and `3`.
```cairo
let keys = ArrayTrait::new();
keys.append('key');
keys.append('deposit');
let values = ArrayTrait::new();
values.append(1);
values.append(2);
values.append(3);
emit_event_syscall(keys, values).unwrap_syscall();
```
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/
cca08c898f0eb3e58797674f20994df0ba641983/corelib/src/starknet/syscalls.cairo#L30)
`library_call`
Syntax
```cairo
extern fn library_call_syscall(
    class_hash: ClassHash, function_selector: felt252, calldata: Span<felt252>
) -> SyscallResult<Span<felt252>> implicits(GasBuiltin, System) nopanic;
```
Description
Calls the requested function in any previously declared class. The class is only used for its logic.
This system call replaces the known delegate call functionality from Ethereum, with the important difference that there is only one contract involved.
Arguments
- _`class_hash`_: The hash of the class you want to use.
- _`function_selector`_: A selector for a function within that class, can be computed with the `selector!` macro.
- _`calldata`_: The calldata.
Return Values
The call response, of type `SyscallResult<Span<felt252>>`.
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/
cca08c898f0eb3e58797674f20994df0ba641983/corelib/src/starknet/syscalls.cairo#L43)
`send_message_to_L1`
Syntax
```cairo
extern fn send_message_to_l1_syscall(
    to_address: felt252, payload: Span<felt252>
```

) -> SyscallResult<()> implicits(GasBuiltin, System) nopanic;
~~~
#### Description
Sends a message to L1.
This system call includes the message parameters as part of the proof’s output and exposes these parameters to the `StarknetCore` contract on L1 once the state update, including the transaction, is received.
For more information, see Starknet’s [messaging mechanism](https://docs.starknet.io/documentation/architecture_and_concepts/Network_Architecture/messaging-mechanism/).
#### Arguments
- _`to_address`_: The recipient’s L1 address.
- _`payload`_: The array containing the message payload.
#### Return Values
None.
#### Example
The following example sends a message whose content is `(1,2)` to the L1 contract whose address is `3423542542364363`.
```cairo
let payload = ArrayTrait::new();
payload.append(1);
payload.append(2);
send_message_to_l1_syscall(payload).unwrap_syscall();
~~~
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/cca08c898f0eb3e58797674f20994df0ba641983/corelib/src/starknet/syscalls.cairo#L51)
`replace_class`
Syntax
```cairo
extern fn replace_class_syscall(
    class_hash: ClassHash
) -> SyscallResult<()> implicits(GasBuiltin, System) nopanic;
```
Description
Once `replace_class` is called, the class of the calling contract (i.e. the contract whose address is returned by `get_contract_address` at the time the syscall is called) will be replaced by the class whose hash is given by the class_hash argument.
> **Note:**
>
> After calling `replace_class`, the code currently executing from the old class will finish running.
>
> The new class will be used from the next transaction onwards or if the contract is called via the `call_contract` syscall in the same transaction (after the replacement).
Arguments

- _`class_hash`_: The hash of the class you want to use as a replacement.
Return Values
None.
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/cca08c898f0eb3e58797674f20994df0ba641983/corelib/src/starknet/syscalls.cairo#L77)
`storage_read`
Syntax
```cairo
extern fn storage_read_syscall(
    address_domain: u32, address: StorageAddress,
) -> SyscallResult<felt252> implicits(GasBuiltin, System) nopanic;
```
Description
Gets the value of a key in the storage of the calling contract.
This system call provides direct access to any possible key in storage, in contrast with `var.read()`, which enables you to read storage variables that are defined explicitly in the contract.
For information on accessing storage by using the storage variables, see [storage variables](https://docs.starknet.io/documentation/architecture_and_concepts/Smart_Contracts/contract-storage/#storage_variables).
Arguments
- _`address_domain`_: The domain of the key, used to separate between different data availability modes. This separation is used in Starknet to offer different data availability modes. Currently, only the onchain mode (where all updates go to L1), indicated by domain `0`, is supported. Other address domains which will be introduced in the future will behave differently in terms of publication (in particular, they will not be posted on L1, creating a tradeoff between cost and security).
- _`address`_: The requested storage address.
Return Values
The value of the key, of type `SyscallResult<felt252>`.
Example
```cairo
use starknet::storage_access::storage_base_address_from_felt252;
...
let storage_address =
storage_base_address_from_felt252(3534535754756246375475423547453)
storage_read_syscall(0, storage_address).unwrap_syscall()
```
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/cca08c898f0eb3e58797674f20994df0ba641983/corelib/src/starknet/syscalls.cairo#L60)
`storage_write`
Syntax
```cairo
extern fn storage_write_syscall(
```

    address_domain: u32, address: StorageAddress, value: felt252
) -> SyscallResult<()> implicits(GasBuiltin, System) nopanic;
~~~
#### Description
Sets the value of a key in the storage of the calling contract.
This system call provides direct access to any possible key in storage, in contrast with `var.write()`, which enables you to write to storage variables that are defined explicitly in the contract.
For information on accessing storage by using the storage variables, see [storage variables](https://docs.starknet.io/documentation/architecture_and_concepts/Smart_Contracts/contract-storage/#storage_variables).
#### Arguments
- _`address_domain`_: The domain of the key, used to separate between different data availability modes. This separation is used in Starknet to offer different data availability modes. Currently, only the onchain mode (where all updates go to L1), indicated by domain `0`, is supported. Other address domains which will be introduced in the future will behave differently in terms of publication (in particular, they will not be posted on L1, creating a tradeoff between cost and security).
- _`address`_: The requested storage address.
- _`value`_: The value to write to the key.
#### Return Values
None.
#### Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/cca08c898f0eb3e58797674f20994df0ba641983/corelib/src/starknet/syscalls.cairo#L70)
## `sha256_process_block`
#### Syntax
```cairo,noplayground
pub extern fn sha256_process_block_syscall(
    state: core::sha256::Sha256StateHandle, input: Box<[u32; 16]>
) -> SyscallResult<core::sha256::Sha256StateHandle> implicits(GasBuiltin, System) nopanic;
~~~
Description
Computes the next SHA-256 state of the input with the given state.
This syscall computes the next SHA-256 state by combining the current `state` with a 512-bit block of `input` data.
Arguments
- _`state`_: The current sha256 state.
- _`input`_: The value to be processed into sha256.
Return Values
Returns a new sha256 state of the `input` data.
Common Library
- [syscalls.cairo](https://github.com/starkware-libs/cairo/blob/3540731e5b0e78f2f5b1a51d3611418121c19e54/corelib/src/starknet/syscalls.cairo#L106)