

Boot Class Hash Table

HashEntry *hash_table;
int hash_size;
int hash_count;
VMLock lock;

...	...
hash(classname)	void *data;
...	...

Class

uintptr_t lock;	
Class *class;	
u1 state;	
u2 access_flags;	
u2 declaring_class;	
u2 enclosing_class;	
u2 enclosing_method;	
u2 inner_access_flags;	
u2 fields_count;	
u2 methods_count;	
u2 interfaces_count;	
u2 inner_class_count;	
u2 constant_pool_count;	
int object_size;	
int method_table_size;	
int imethod_table_size;	
int initing_tid;	
char *name;	
char *signature;	
char *source_file_name;	
Class *super;	
Class **interfaces;	
MethodBlock **method_table;	
ITableEntry *imethod_table;	
char *bootstrap_methods;	
ExtraAttributes *extra_attributes;	
ConstantPool constant_pool;	
int dimension;	u2 *inner_classes;
	FieldBlock *fields;
	MethodBlock *methods;
Class *element_class;	int refs_offsets_size;
	RefsOffsetsEntry *refs_offsets_table;

} class header

FieldBlock

Class *class;
char *name;
char *type;
char *signature;
u2 access_flags;
u2 constant;
char data[8]; or offset

MethodBlock

Class *clas;	
char *name;	
char *type;	
char *signature;	
u1 state;	
u1 flags;	
u2 access_flags;	
u2 max_stack;	
u2 max_locals;	
u2 args_count;	
u2 throw_table_size;	
u2 *throw_table;	
void *code;	
int code_size;	
int method_table_index;	
u2 exception_table_size;	char *simple_sig;
u2 line_no_table_size;	
ExceptionTableEntry *exception_table;	int native_extra_arg;
LineNoTableEntry *line_no_table;	NativeMethod native_invoker;