```
Error in compiled code, beam file damaged c
                Unexpected op code N
                Cannot allocate <N> bytes of memory (of type "<T>"
                                                                     run out of memory
                                                                         The kernel/stdlib applications are damaged
                "Module Name undefined" | "Function Name undefined"
                function Name:start/2"
                                                                         The start script is damaged
                                                                                file descriptors for sockets exceed 1024 (Unix
                                                                                The limit on file-descriptors in some Unix flavors can be set to over 10
                Driver_select called with too large file descriptor N
                                                                                simultaneously by Erlang (due to limitations in the Unix select call). The
                Received SIGUSR1
                                              Unix only
                                                                       usually (Who = application_controller, Why = shutdown).
                                                                                                                                              usually node name is already in use
Slogan
                                                                                                                         It is always due to some kind of failure in an applicati
                                                                                                                         written" one. Looking at the error log for your applical
                Kernel pid terminated (Who) (Exit-reason)
                                                                      This message comes from the Erlang code and no
                                                         The primitive Erlang boot sequence was terminated,
                                                         script has errors or cannot be read.
                "Init terminating in do_boot ()"
                                                         This is usually a configuration error - the system may boot parameter or with a boot script from the wrong
                                                               One of the kernel processes could not start.
                                                               Config files are in their correct location and that the cc
                                                               damaged.
                "Could not start kernel pid (Who) ()"
                                                               Usually there are also messages written to the control
                                                               explaining what's wrong.
Number of atoms
Memory information
                                     hash_table
Internal table information
                                     index_table
                           erlang:system_info(allocated_areas).
Allocated areas
                   erlang:system_info({allocator, <A>}).
Allocator
                                             Scheduled - The process was scheduled to run but not currently running ("in the run queue").
                                             Waiting - The process was waiting for something (in receive).
                                             Running - The process was currently running. If the BIF erlang:halt/1 was called, this was the process calling it.
                                State
                                             Exiting - The process was on its way to exit.
                                             Garbing - This is bad luck, the process was garbage collecting when the crash dump was written, the rest of the information for this proces
                                             Suspended - The process is suspended, either by the BIF erlang:suspend_process/1 or because it is trying to write to a busy port.
                                Registered name
                                Spawned as
                                Last scheduled in for | Current call
                                Spawned by
                                                     The parent of the process
                                Started
                                Message queue length
                                Number of heap fragments
Process information
                                                              Size of fragmented heap data. This is data either created by messages being sent to the process or by the Erlang BIFs.
                                Heap fragment data
                                                May also contain ports.
                                Link list
                                                 If process monitoring is used, this field also tells in which direction the monitoring is in effect, i.e., a link being "to" a
                                                 "from" a process tells you that the other process was monitoring the current one.
                                Reductions
                                                     The size of the stack and heap (they share memory segment)
                                Stack+heap
                                Heap unused, OldHeap unused
                                              If the system uses shared heap, the fields Stack+heap, OldHeap, Heap unused and OldHeap unused do not exist. Instead this field pre
                                Stack
                                Program counter
                                                  The size of the "old heap". The Erlang virtual machine uses generational garbage collection with two generations. There is one heap fo
                                                  (which is almost always correct) is that data that survive two garbage collections can be "tenured" to a heap more seldom garbage collections
                                                  sum of the heaps and stack together constitute most of the process's allocated memory.
                                OldHeap
                           This section lists the open ports, their owners, any linked processed, and the name of their driver or external process.
Port information
                     owner
                                      This occurs if the table is a hash table, i.e. if it is not an ordered_set.
                     buckets
                                                                  This occurs only if the table is an ordered_set. (The number of elements is the same as the number of objects in to
ETS tables
                     Ordered set (AVL tree), Elements
                                      The number of objects in the table
                     Objects
                                     The number of words (usually 4 bytes/word) allocated to data in the table.
                     Words
                Message
Timers
                Time left
Distribution information
Loaded module information
Fun information
Process Data
               Note that the last created atom is printed first.
```

Erlang Crash Dumps

**Atoms**