Proposed Solution: Examining Memory Usage and Compile Time

Automated localization of memory mismanagement
- Built using C Intermediate Language (CIL)
- Combines dataflow and alias analysis
- Configured using annotation templates

Function | Total uses out of 46371 SLOC |
--- | --- |
`sys_malloc` | 135 |
`sysmsg_take_data` | 23 |
`sfree` | 124 |
`SOS_MESSAGE_RELEASE` | 116 |
Other common memory operations | 54 |

Memorie usage in SOS operating system

Memorie usage in SOS user applications

Evaluation on the SOS operating system for sensor networks

Limitations
- Analysis requires some annotations
- Annotations kept outside of code
- Small set provides significant benefits
- Limited accuracy in alias analysis
- Checker can use any external alias analysis engine
- Analysis has no concept of state

Future work
- Extend verification with a model of the application
- Formal type system for inference
- Integration with link time and run time checking

Focus on locating memory mismanagement in sensor networks
- Models used for memory apply to other resource management problems
- Mismanaged memory leads to significant problems
- Memory leaks rapidly exhaust limited memory available to node
- No MMU protection so dereferencing a dangling pointer can crash a node or corrupt state in other applications

Proposed Solution: Examining Memory Usage and Compile Time

Compile time feedback for application developers
- Incorrect memory usage displayed as compiler warnings
- Checker integrated with build process