WEP4P1

Backported & Under Developing Items in amorita Private Repository

Akio Morita(SAD committer in SuperKEKB commissioning group),
KEK, 1-1, Oho, Tsukuba, Ibaraki 305-0801, Japan

Introduction

amorita private repository is my private repository forked from KEKB SAD main repository. It is used for my SAD environment and new feature development. The bugs found in my daily works are fixed on this repository.

Both new feature developed on this repository and fixed errata are back-ported into KEKB SAD main trunk if these are required for SuperKEKB operation.

In this poster, major back-ported items, under developing items and future development ideas are presented.

CAUTION:
This item list is not complete and does not contain "How to use". It is provided as hints for reading source codes.

Note:
New developed features are not back-ported into k64 branch, because I'm not a k64 branch developer. Therefore, SuperKEKB operation software framework CAN NOT work on k64 branch.

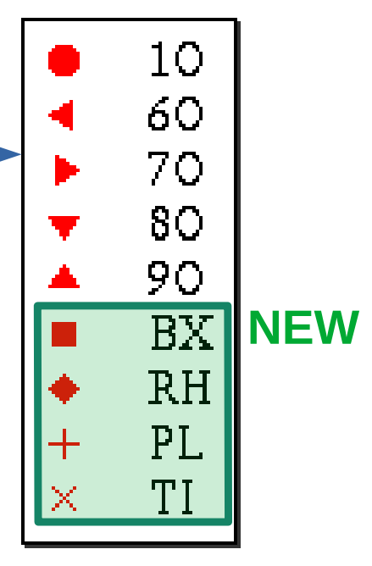
Under Developing Items

* New builtin function table framework
 - Provide unified & efficient table dispatch for builtin functions
* Builtin math function generator
 - Generate wrapper code for builtin math function from configuration file
 - Generated wrapper code supports nesting List map action
* Fix incorrect EvenQ/OddQ results for example following corner cases:
 - EvenQ[Infinity], OddQ[Infinity], OddQ[2^64]
* Fix incorrect Bessel function family(BesselJ, BesselY, BesselI, BesselK)
 - Fix incorrect result on negative real axis
 - Fix incorrect branch selection in complex plan
* Address independent inter-process shared memory(Shared[] updates)
 - Introduce address independent object dump format
* Portable abstraction layer for SAD object container manipulation
  For example:
  - sad_strlen - String length of SAD String object
  - sad_cstr - C const char* pointer of SAD String object contents
  - sad_list_size - Size of SAD list container
 - Abstracted SAD container iterater
* Portable abstraction layer for SAD interpreter stack
* Replacing on-stack list construction with on-heap construction in order to avoid stack corruption
* Refacotring "**un-documented**" & "**un-structured**" internal data structure
 - Introduce well-defined structured data type
 - Introduce type-safe accessor interfaces
* Replacing local array allocation by using 64bit un-safe heap allocator with **allocable** array
* Improve inter-process parallel contexts
 - Replace wait(2) with wait(4) to avoid catching background sub-process instead of co-worker.

Back-ported Items

* Dynamic loader
 - DynamicLink[] - Load shared object
 - DynamicCall[] - Call SADScript function interface via dlsym(3)
* FFICall extention module
 - Call foreign function interface via libffi
* LAPACK extension module
 - Faster replacement of LinearSolve[], SingularValues[], EigenSystem[]
* FFTW extension module
 - Faster replacement of Fourier[], InverseFourier[]
* PkgConfig & PkgHook interface
 - Provide customize point for package functions
 - Provide customize hook evaluating after package loading
* New EPICS channel access backend (**WEP4P2**)
* UTF-8 Support
 - \u#### - UTF-8 character(U+####) in String literal
 - \U######## - UTF-8 character(U+########) in String literal
 - Provide character escape mode switching for Tkinter by using BOM/MOB character in order to pass UTF-8 string(including CJK characters) to Tkinter.
* LOSSMAP extension
 - Provide particle loss position(trun & beamline-element number)
 - Easy to build particle loss map generation by using multi-turn tracking
* Signal support
 - Provide SigPending[], SigSuspend[], SigProcMask[], SigAction[]
* Race free Shread[]
 - Provide race free reader/writer lock for inter-process shared memory
* To/FromDateString[] Format option extension
 - Support str[fp]time(3) format string via **Format->"+***FORMAT_STRING***"** option

* Floor[], Ceiling[], Round[] updates
 - Fix incorrect floating point rounding for example following corner cases:
  + Round[.49999999999999994], Round[9007199254740991]
  + Floor[.9999999999999999], Ceiling[-.9999999999999999]
* LogGamma[]/Gamma[] updates
 - Fix singular point location on negative real axis
 - Fix continuity of LogGamma[] for derivative
* TCPShutdown[]
 - Shutdown TCP stream socket by using shutdown(2)
* TCPAccept[] improvements
 - Provide non-blocking accept(2)
 - Provide accept(2) for multiple listen sockets via socket list.
* TCPServer[] wrapper
 - Make TCP server sockets(List) for multi stack network
  (for example: IPv4 / IPv6 dual stack system)
* PollUnit[]
 - poll(2) wrapper same as SelectUnit[]
* FilePipe[]
 - Pipe[] replacement for **foreground** sub-process
* FileLock[]
 - Provide shared/exclusive(F_RDLCK/F_WELCK) file lock by using fcntl(2)
* Chmod[] - chmod(2) wrapper
* Umask[] - umask(2) wrapper
* PlotSymbol extension for CanvasDrawer
* Portable mmap(2) helper function for Frotran shared array allocation
* New USES/OMITS feature selection directive for build framework
 - USES += EPICSCA - Offer EPICS CA interface
 - USES += Tkinter - Offer Tkinter
* New type safe container accessor for Fortran/C coding



Future Development Ideas

* Introduce per element aperture
 - Arbitrary aperture definition via shape function
 - Aperture code JIT for typical aperture shape
* Reconstruction tracking code driver framework
 - Unify call interface prototype for element type implementation codes.
 - Replace "Computed GOTO Statement" based element type dispatch with pointer dispatch.
 - Replace element parameter based dispatch with pre-selection of optimized element map codes.
  (Reduce runtime per element conditional branch by pre-building element map pointer array)
 - Intrdocude dynamic extendable element type defintion.
* Replace process based parallel tracking with multi thread based code
 - Introduce inter-thread barrier for synchronizing in order to support particle interaction and reduction operator.
  (for example: wake potential, luminosity calcuration among parallel context)
* Fine grading parallel tracking for DynamicApertureSurvey[]
 - Reduce idling co-worker process by using inter-process core board to take un-processed task.
 - Improve task dispatch algorithm.
 - It WOULD improve latency
* Improve DynamicApertureSurvey[] performance.
 - Improve border detection algorithm to reduce evaluation points.
  + Interlace scaning
  + Use nearby momentum information
 - Improve for lifetime evaluation
  + Evaluate around aperture ellipse for Touschek lifetime evaluation
* Introduce type-safe & container format independent abstraction layer
 - k64 branch implementation is not clearly splited container internal data type from high level codes.

Current Development Status of KEK Repository

* KEKB SAD repository is hosted SADNAS.
 - Anonymous access is provided for KEK network.
  http://afsad1.kek.jp/SAD/svnroot/
 - No official distribution site for internet, because of lack of hostring & content management resouces.

* 22 committers are registered.
 - 1 active committer for MAIN trunk.

* MAIN trunk is maintained for SuperKEKB.
 - SAD is operated on FreeBSD 12-STABLE(amd64).
 - Keeping backward compatibility for SuperKEKB operating software strack.

Other Informations

* SAD home page
  http://acc-physics.kek.jp/SAD/
* Conference room in home page
  (User community support by volunteer)
  http://acc-physics.kek.jp/cgi-bin/SAD2/wforum.cgi