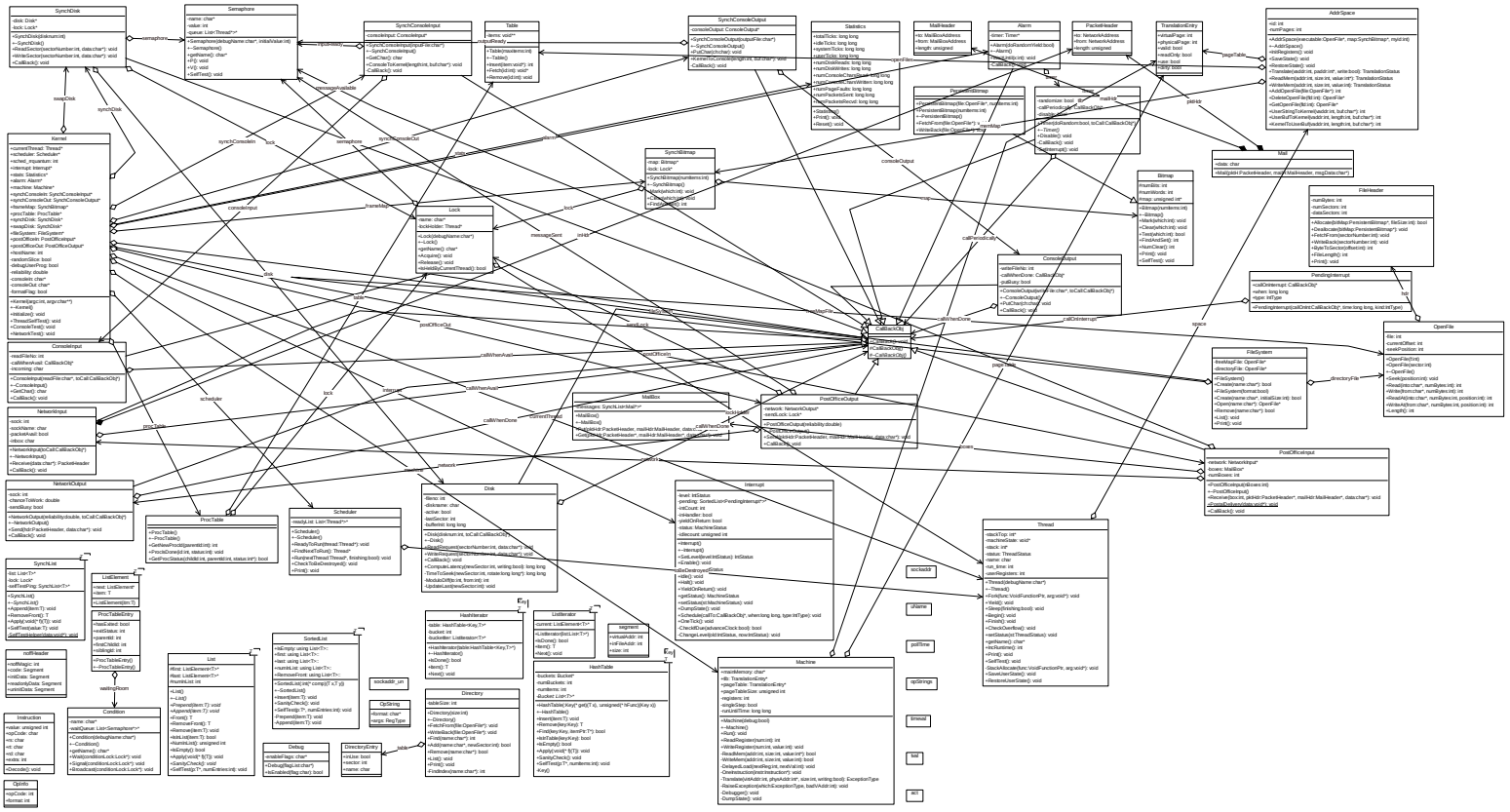
SynchDisk
-disk: Disk*
-lock: Lock*
+SynchDisk(diskname:char)
+~SynchDisk()
+ReadSector(sectorNumber:int, data:char*): void
+WriteSector(sectorNumber:int, data:char*): void
+CallBack(): void
Semaphore
-name: char*
-value: int
-queue: List<Thread>*
+Semaphore(debugName:char*, initialValue:int)
+~Semaphore()
+getName(): char*
+P(): void
+V(): void
+SelfTest(): void
SynchConsoleInput
-consoleInput: ConsoleInput*
+SynchConsoleInput(inputFile:char*)
+~SynchConsoleInput()
+GetChar(): char
+GetMaxLineLength(int, int, buf:char*): void
-CallBack(): void
Table
-items: void*
-inUsedNearly: int
+Table(maxItems:int)
+~Table()
+Insert(item:void*): int
+Fetch(id:int): void*
+Remove(id:int): void
SynchConsoleOutput
-consoleOutput: ConsoleOutput*
+SynchConsoleOutput(outputFile:char*)
+~SynchConsoleOutput()
+PutChar(ch:char): void
+PutString(buffer:char*, bufLength:int, buf:char*): void
-CallBack(): void
Statistics
+totalTicks: long long
+idleTicks: long long
+systemTicks: long long
+userTicks: long long
+numDiskReads: long long
+numDiskWrites: long long
+numConsoleCharsRead: long long
+numConsoleCharsWritten: long long
+numPageFaults: long long
+numPacketsSent: long long
+numPacketsRecvd: long long
+Statistics()
+Print(): void
+Reset(): void
MailHeader
+to: MailBoxAddress
+from: MailBoxAddress
+length: unsigned
Alarm
-timer: Timer*
+Alarm(int pRandom:Void bool)
+~Alarm()
+CallBack(int): void
PacketHeader
+to: NetworkAddress
+from: NetworkAddress
+length: unsigned
TranslationEntry
+virtualPage: int
+physicalPage: int
+valid: bool
+readOnly: bool
+use: bool
+dirty: bool
AddrSpace
-vol: int
-numPages: int
+AddrSpace(executable:OpenFile*, map:SynchBitmap*, mpid:int)
+~AddrSpace()
+InitRegisters(): void
+SaveState(): void
+RestoreState(): void
+Translate(vaddr:int, paddr:int*, write:bool): TranslateStatus
+ReadMem(vaddr:int, size:int, value:int*): TranslateStatus
+WriteMem(vaddr:int, size:int, value:int): TranslateStatus
+AddOpenFile(file:OpenFile*): int
+DeleteOpenFile(fid:int): OpenFile*
+GetOpenFile(fid:int): OpenFile*
+UserGetNgTo(name:char*, buf:char*): int
+UserBufToKernel(vaddr:int, length:int, buf:char*): int
+KernelToUserBuf(vaddr:int, length:int, buf:char*): int
Kernel
-currentThread: Thread*
-scheduler: Scheduler*
-interrupt, stopIntr: int*
-interrupt: Interrupt*
-stats: Statistics*
-alarm: Alarm*
-machine: Machine*
-synchConsoleIn: SynchConsoleInput*
-synchConsoleOut: SynchConsoleOutput*
-frameMap: SynchBitmap*
-procTable: ProcTable*
-synchDisk: SynchDisk*
-swapDisk: SynchDisk*
-fileSystem: FileSystem*
-postOfficeIn: PostOfficeInput*
-postOfficeOut: PostOfficeOutput*
-hostname: int
-randomSlice: bool
-debugUserProg: bool
-reliability: double
-consoleIn: char*
-consoleOut: char*
-formatFlag: bool
+Kernel(argc:int, argv:char**)
+~Kernel()
+Initialize(): void
+ThreadSelfTest(): void
+ConsoleTest(): void
+NetworkTest(): void
Mail
+data: char
+Mail(pktHdr:PacketHeader, mailHdr:MailHeader, msgData:char*)
SynchBitmap
-map: Bitmap*
-lock: Lock*
+SynchBitmap(numItems:int)
+~SynchBitmap()
+getNumFreeBits(int): void
+FindAndSet(int*): bool
+Find(int*): int
Lock
-name: char*
-lockHolder: Thread*
+Lock(debugName:char*)
+~Lock()
+getName(): char*
+Acquire(): void
+Release(): void
+IsHeldByCurrentThread(): bool
Bitmap
-numBits: int
-numWordsInUse: int
-map: unsigned int*
+Bitmap(numItems:int)
+~Bitmap()
+MarkAll(int,int): void
+Clear(int,int): void
+Test(int,int): bool
+FindAndSet(int*): int
+NumClear(): int
+Print(): void
+SelfTest(): void
FileHeader
-numBytes: int
-numSectors: int
-dataSectors: int
+AllocateBitMap(PersistentBitmap*, fileSize:int): bool
+DeallocateBitMap(PersistentBitmap*): void
+FetchFrom(sectorNumber:int): void
+WriteBack(sectorNumber:int): void
+ByteToSector(offset:int): int
+FileLength(): int
+Print(): void
ConsoleInput
-readFileNo: int
-callBackObj: CallBackObj*
-incoming: char
+ConsoleInput(readFile:char*, toCallCallBackObj*)
+~ConsoleInput()
+GetChar(): char
+CallBack(): void
ConsoleOutput
-writeFileNo: int
-callBackDone: CallBackObj*
-putBusy: bool
+ConsoleOutput(writeFile:char*, toCallCallBackObj*)
+~ConsoleOutput()
+PutChar(ch:char): void
+CallBack(): void
PendingInterrupt
+callOnInterrupt: CallBackObj*
+when: long long
+type: IntType
+PendingInterrupt(callOnInt:CallBackObj*, time:long long, kind:IntType)
NetworkInput
-sock: int
-socketName: char
-packetAvail: bool
-mbox: char
+NetworkInput(toCallCallBackObj*)
+~NetworkInput()
+Receive(data:char*): PacketHeader
+CallBack(): void
NetworkOutput
-sock: int
-charsorTotWork: double
-sendBusy: bool
+NetworkOutput(availability:double, toCallCallBackObj*)
+~NetworkOutput()
+SendPacket(header, data:char*): void
+CallBack(): void
MailBox
-messages: SynchList<MailH*>
+MailBox()
+~MailBox()
+PutMsg(pktHdr:PacketHeader, mailHdr:MailHeader, data:char*)
+GetMsg(pktHdr:PacketHeader*, mailHdr:MailHeader*, data:char*)
PostOfficeOutput
-network: NetworkOutput*
-sendLock: Lock*
+PostOfficeOutput(reliability:double)
+~PostOfficeOutput()
+SendMail(destPacketHeader, mailHdr:MailHeader, data:char*): void
+CallBack(): void
OpenFile
-file: int
-currentOffset: int
-seekPosition: int
+OpenFile(int)
+OpenFile(sector:int)
+~OpenFile()
+Seek(position:int): void
+Read(into:char*, numBytes:int): int
+Write(from:char*, numBytes:int): int
+ReadAt(into:char*, numBytes:int, position:int): int
+WriteAt(from:char*, numBytes:int, position:int): int
+Length(): int
FileSystem
-firstMapFile: OpenFile*
-directoryFile: OpenFile*
+FileSystem()
+Create(name:char*): bool
+FileSystem(format:bool)
+Create(name:char*, initialSize:int): bool
+Open(name:char*): OpenFile*
+Remove(name:char*): bool
+List(): void
+Print(): void
PostOfficeInput
-network: NetworkInput*
-mailBox: MailBox*
-numBoxes: int
+PostOfficeInput(numBoxes:int)
+~PostOfficeInput()
+Receive(pktHdr, pktHdr:PacketHeader*, mailHdr:MailHeader*, data:char*): void
+PostDelivery(data:void*): void
+CallBack(): void
ProcTable
+ProcTable()
+~ProcTable()
+GetNewProcID(parentID:int): int
+ProcDone(pid:int, status:int): void
+GetProcStatus(childid:int, parentid:int, status:int*): bool
Scheduler
-readyList: List<Thread>*
+Scheduler()
+~Scheduler()
+ReadyToRun(thread:Thread*): void
+FindNextToRun(): Thread*
+Run(nextThread:Thread*, finishing:bool): void
+CheckToBeDestroyed(): void
+Print(): void
Disk
-fileno: int
-diskname: char
-active: bool
-lastSector: int
-bufferInit: long long
+Disk(pfileno:int, toCallCallBackObj)
+~Disk()
+ReadSector(sectorNumber:int, data:char*): void
+WriteSector(sectorNumber:int, data:char*): void
+CallBack(): void
+ComputeLatency(newSector:int, writing:bool): long long
-TimeToSeek(newSector:int, rotate:long long*): long long
-ModulO(dfp:int, from:int): int
+UpdateLast(newSector:int): void
Interrupt
-level: IntStatus
-pending: SortedList<PendingInterrupt>*
-intCount: int
-inHandler: bool
-yieldOnReturn: bool
-status: MachineStatus
-tReturnt: unsigned int
+Interrupt()
+~Interrupt()
+SetLevel(level:IntStatus): IntStatus
+Enable(): void
+IsEnabled(): IntStatus
+Idle(): void
+Halt(): void
+YieldOnReturn(): void
+getStatus(): MachineStatus
+setStatus(st:MachineStatus): void
+DumpState(): void
+Schedule(callToCallBackObj*, when:long long, type:IntType): void
+OneTick(): void
-CheckIfDue(advanceClock:bool): bool
-ChangeLevel(toIntStatus, newIntStatus): void
Thread
-stackTop: int*
-machineState: void*
-stack: int*
-status: ThreadStatus
-name: char
-tot_time: int
-userRegisters: int
+Thread(debugName:char*)
+~Thread()
+Fork(func:VoidFunctionPtr, arg:void*): void
+Yield(): void
+Sleep(finishing:bool): void
+Begin(): void
+Finish(): void
+CheckOverflow(): void
+setStatus(st:ThreadStatus): void
+getName(): char*
+IncRuntime(): int
+Print(): void
+SelfTest(): void
-StackAllocate(func:VoidFunctionPtr, arg:void*): void
+SaveUserState(): void
+RestoreUserState(): void
socketdir
sfname
pofiltne
opStrings
timeout
bid
act
NetworkHeader
+netMagic: int
+code: Segment
+initData: Segment
+readonlyData: Segment
+uninitData: Segment
Instruction
+value: unsigned int
+opCode: char
+rs: char
+rt: char
+rd: char
+extra: int
+Decode(): void
OpInfo
+opCode: int
+format: int
SynchList
-list: List<T>*
-lock: Lock*
-setFirstFreq: SynchList<T>*
+SynchList()
+~SynchList()
+Append(item:T): void
+RemoveFront(): T
+Apply(void*, t(T)): void
+SelfTest(value:T): void
-GetFreeNode(data:void*): void
ListElement
+next: ListElement*
+item: T
+ListElement(item:T)
ProcTableEntry
+hasExited: bool
+exitStatus: int
+parentid: int
+lastChildid: int
+oldlargid: int
+ProcTableEntry()
+~ProcTableEntry()
waitingRoom
Condition
-name: char*
-waitQueue: List<Semaphore>*
+Condition(debugName:char*)
+~Condition()
+getName(): char*
+Wait(conditionLock:Lock*): void
+Signal(conditionLock:Lock*): void
+Broadcast(conditionLock:Lock*): void
List
-fRoot: ListElement<T>*
-fLast: ListElement<T>*
-fNumInList: int
+List()
+~List()
+Prepend(item:T): void
+Append(item:T): void
+Fcnt(): T
+RemoveFcnt(): T
+Remove(item:T): void
+IsInList(item:T): bool
+NumInList(): unsigned int
+IsEmpty(): bool
+Apply(void*, t(T)): void
+SanityCheck(): void
+SelfTest(p:T*, numEntries:int): void
SortedList
+IsEmpty: using List<T>;
+first: using List<T>;
+last: using List<T>;
+numInList: using List<T>;
+RemoveFront: using List<T>;
+SortedList(int(*cmp)(T a,T y))
+~SortedList()
+Insert(item:T): void
+SanityCheck(): void
+SelfTest(p:T*, numEntries:int): void
-Prepend(item:T): void
-Append(item:T): void
Debug
-enabledFlags: char*
+DebugFlag(list:char*)
+IsEnabled(flag:char): bool
DirectoryEntry
-inUse: bool
-inector: int
-name: char
sockaddr_un
OpString
-fformat: char*
-args: ArgType
HashIterator
-table: HashTable<Key,T>*
-bucket: int
-bucketNm: ListIterator<T>*
+HashIterator(table:HashTable<Key,T>*)
+~HashIterator()
+IsDone(): bool
+Next(): T
+Next(): void
ListIterator
-current: ListElement<T>*
+ListIterator(list:List<T>*)
+IsDone(): bool
+Next(): T
+Next(): void
segment
+virtualAddr: int
+inFileAddr: int
+size: int
Directory
-tableSize: int
+Directory(size:int)
+~Directory()
+FetchFrom(file:OpenFile*): void
+WriteBackTo(file:OpenFile*): void
+Find(name:char*): int
+Add(name:char*, newSector:int): bool
+Remove(name:char*): bool
+List(): void
+Print(): void
+FindIndex(name:char*): int
HashTable
-buckets: Bucket*
-numBuckets: int
-numItems: int
-Bucket List<T>*
+HashTable(key(Key*):pe(T,t), unsigned(*)(Key,n))
+~HashTable()
+insert(item:T): void
+Remove(keyKey):T
+Find(keyKey,itemPtr:T*): bool
+isInTable(keyKey): bool
+IsEmpty(): bool
+Apply(void*, t(T)): void
+SanityCheck(): void
+SelfTest(p:T*, numItems:int): void
+Key()
Machine
+mainMemory: char*
+tlb: TranslationEntry*
+pageTable: TranslationEntry*
+pageTableSize: unsigned int
-registers: int
-singleStep: bool
-runUntilTime: long long
+Machine(debug:bool)
+~Machine()
+Run(): void
+ReadRegister(num:int): int
+WriteRegister(num:int, value:int): void
-ReadMem(vaddr:int, size:int, value:int*): bool
-WriteMem(vaddr:int, size:int, value:int): bool
-DelayedLoad(nextReg:int, nextVal:int): void
-OneInstruction(instr:Instruction*): void
-Translate(virtAddr:int, physAddr:int*, size:int, writing:bool): ExceptionType
-RaiseException(which:ExceptionType, badVAddr:int): void
-Debugger(): void
-DumpState(): void