*** BIG FAT WARNING ***
The kvm module is currently in EXPERIMENTAL state for s390. This means that the interface to the module is not yet considered to remain stable. Thus, be prepared that we keep breaking your userspace application and guest compatibility over and over again until we feel happy with the result. Make sure your guest kernel, your host kernel, and your userspace launcher are in a consistent state.

This Documentation describes the unique ioctl calls to /dev/kvm, the resulting kvm-vm file descriptors, and the kvm-vcpu file descriptors that differ from x86.

1. ioctl calls to /dev/kvm
KVM does support the following ioctls on s390 that are common with other architectures and do behave the same:
KVM_GET_API_VERSION
KVM_CREATE_VM (*) see note
KVM_CHECK_EXTENSION
KVM_GET_VCPU_MMAP_SIZE

Notes:
* KVM_CREATE_VM may fail on s390, if the calling process has multiple threads and has not called KVM_S390_ENABLE_SIE before.

In addition, on s390 the following architecture specific ioctls are supported:
ioctl: KVM_S390_ENABLE_SIE
args: none
see also: include/linux/kvm.h
This call causes the kernel to switch on PGSTE in the user page table. This operation is needed in order to run a virtual machine, and it requires the calling process to be single-threaded. Note that the first call to KVM_CREATE_VM will implicitly try to switch on PGSTE if the user process has not called KVM_S390_ENABLE_SIE before. User processes that want to launch multiple threads before creating a virtual machine have to call KVM_S390_ENABLE_SIE, or will observe an error calling KVM_CREATE_VM. Switching on PGSTE is a one-time operation, is not reversible, and will persist over the entire lifetime of the calling process. It does not have any user-visible effect other than a small performance penalty.

2. ioctl calls to the kvm-vm file descriptor
KVM does support the following ioctls on s390 that are common with other architectures and do behave the same:
KVM_CREATE_VCPU
KVM_SET_USER_MEMORY_REGION (*) see note
KVM_GET_DIRTY_LOG (**) see note

Notes:
* kvm does only allow exactly one memory slot on s390, which has to start at guest absolute address zero and at a user address that is aligned on any page boundary. This hardware "limitation" allows us to have a few unique optimizations. The memory slot doesn't have to be filled with memory actually, it may contain sparse holes. That said, with different user memory layout this does still allow a large flexibility when doing the guest memory setup.
** KVM_GET_DIRTY_LOG doesn't work properly yet. The user will receive an empty log. This ioctl call is only needed for guest migration, and we intend to implement this one in the future.

In addition, on s390 the following architecture specific ioctls for the kvm-vm file descriptor are supported:
ioctl: KVM_S390_INTERRUPT
args: struct kvm_s390_interrupt *
see also: include/linux/kvm.h
This ioctl is used to submit a floating interrupt for a virtual machine. Floating interrupts may be delivered to any virtual cpu in the configuration. Only some interrupt types defined in include/linux/kvm.h make sense when submitted as floating interrupts. The following interrupts are not considered to be useful as floating interrupts, and a call to inject them will result in -EINVAL error code: program interrupts and interprocessor signals. Valid floating interrupts are:
KVM_S390_INT_VIRTIO
KVM_S390_INT_SERVICE

3. ioctl calls to the kvm-vcpu file descriptor
KVM does support the following ioctls on s390 that are common with other architectures and do behave the same:
KVM_RUN
KVM_GET_REGS
KVM_SET_REGS
KVM_GET_SREGS
KVM_SET_SREGS
KVM_GET_FPU
KVM_SET_FPU

In addition, on s390 the following architecture specific ioctls for the kvm-vcpu file descriptor are supported:
ioctl: KVM_S390_INTERRUPT
args: struct kvm_s390_interrupt *
see also: include/linux/kvm.h
This ioctl is used to submit an interrupt for a specific virtual cpu. Only some interrupt types defined in include/linux/kvm.h make sense when submitted for a specific cpu. The following interrupts are not considered to be useful, and a call to inject them will result in -EINVAL error code: service processor calls and virtio interrupts. Valid interrupt types are:
KVM_S390_PROGRAM_INT
KVM_S390_SIGP_STOP
KVM_S390_RESTART
KVM_S390_SIGP_SET_PREFIX
KVM_S390_INT_EMERGENCY

ioctl: KVM_S390_STORE_STATUS
args: unsigned long
see also: include/linux/kvm.h
This ioctl stores the state of the cpu at the guest real address given as argument, unless one of the following values defined in include/linux/kvm.h is given as argument:
KVM_S390_STORE_STATUS_NOADDR - the CPU stores its status to the save area in absolute lowcore as defined by the principles of operation
KVM_S390_STORE_STATUS_PREFIXED - the CPU stores its status to the save area in its prefix page just like the dump tool that comes with zipl. This is useful to create a system dump for use with lkcdutils or crash.

ioctl: KVM_S390_SET_INITIAL_PSW

args: struct kvm_s390_psw *
see also: include/linux/kvm.h
This ioctl can be used to set the processor status word (psw) of a stopped cpu prior to running it with KVM_RUN. Note that this call is not required to modify the psw during sie intercepts that fall back to userspace because struct kvm_run does contain the psw, and this value is evaluated during reentry of KVM_RUN after the intercept exit was recognized.

ioctl: KVM_S390_INITIAL_RESET
args: none
see also: include/linux/kvm.h
This ioctl can be used to perform an initial cpu reset as defined by the principles of operation. The target cpu has to be in stopped state.