FLIC (floating interrupt controller)

FLIC handles floating (non per-cpu) interrupts, i.e. I/O, service and some machine check interruptions. All interrupts are stored in a per-vm list of pending interrupts. FLIC performs operations on this list.

Only one FLIC instance may be instantiated.

FLIC provides support to - add interrupts (KVM_DEV_FLIC_ENQUEUE) - inspect currently pending interrupts (KVM_FLIC_GET_ALL_IRQS) - purge all pending floating interrupts (KVM_DEV_FLIC_CLEAR_IRQS) - purge one pending floating I/O interrupt (KVM_DEV_FLIC_CLEAR_IO_IRQ) - enable/disable for the guest transparent async page faults - register and modify adapter interrupt sources (KVM_DEV_FLIC_ADAPTER_*) - modify AIS (adapter-interruption-suppression) mode state (KVM_DEV_FLIC_AISM) - inject adapter interrupts on a specified adapter (KVM_DEV_FLIC_AIRQ_INJECT) - get/set all AIS mode states (KVM_DEV_FLIC_AISM_ALL)

Groups:

KVM_DEV_FLIC_ENQUEUE

Passes a buffer and length into the kernel which are then injected into the list of pending interrupts. attr->addr contains the pointer to the buffer and attr->attr contains the length of the buffer. The format of the data structure kvm_s390_irq as it is copied from userspace is defined in usr/include/linux/kvm.h.

KVM_DEV_FLIC_GET_ALL_IRQS

Copies all floating interrupts into a buffer provided by userspace. When the buffer is too small it returns -ENOMEM, which is the indication for userspace to try again with a bigger buffer.

-ENOBUFS is returned when the allocation of a kernelspace buffer has failed.

-EFAULT is returned when copying data to userspace failed. All interrupts remain pending, i.e. are not deleted from the list of currently pending interrupts. attr->addr contains the userspace address of the buffer into which all interrupt data will be copied. attr->attr contains the size of the buffer in bytes.

KVM_DEV_FLIC_CLEAR_IRQS

Simply deletes all elements from the list of currently pending floating interrupts. No interrupts are injected into the guest.

KVM_DEV_FLIC_CLEAR_IO_IRQ

Deletes one (if any) I/O interrupt for a subchannel identified by the subsystem identification word passed via the buffer specified by attr->addr (address) and attr->attr (length).

KVM_DEV_FLIC_APF_ENABLE

Enables async page faults for the guest. So in case of a major page fault the host is allowed to handle this async and continues the guest.

KVM_DEV_FLIC_APF_DISABLE_WAIT

Disables async page faults for the guest and waits until already pending async page faults are done. This is necessary to trigger a completion interrupt for every init interrupt before migrating the interrupt list.

KVM_DEV_FLIC_ADAPTER_REGISTER

Register an I/O adapter interrupt source. Takes a kvm_s390_io_adapter describing the adapter to register:

```
struct kvm_s390_io_adapter {
        __u32 id;
        __u8 isc;
        __u8 maskable;
        __u8 swap;
        __u8 flags;
};
```

id contains the unique id for the adapter, isc the I/O interruption subclass to use, maskable whether this adapter may be masked (interrupts turned off), swap whether the indicators need to be byte swapped, and flags contains further characteristics of the adapter.

Currently defined values for 'flags' are:

- KVM_S390_ADAPTER_SUPPRESSIBLE: adapter is subject to AIS (adapter-interrupt-suppression) facility. This flag only has an effect if the AIS capability is enabled.

Unknown flag values are ignored.

KVM_DEV_FLIC_ADAPTER_MODIFY

Modifies attributes of an existing I/O adapter interrupt source. Takes a kvm_s390_io_adapter_req specifying the adapter and the operation:

```
struct kvm_s390_io_adapter_req {
        __u32 id;
        __u8 type;
        __u8 mask;
        __u16 pad0;
        __u64 addr;
};
```

id specifies the adapter and type the operation. The supported operations are:

KVM_S390_IO_ADAPTER_MASK
: mask or unmask the adapter, as specified in mask

KVM_S390_IO_ADAPTER_MAP
: This is now a no-op. The mapping is purely done by the irq route.

KVM_S390_IO_ADAPTER_UNMAP
: This is now a no-op. The mapping is purely done by the irq route.

KVM_DEV_FLIC_AISM

modify the adapter-interruption-suppression mode for a given isc if the AIS capability is enabled. Takes a kvm_s390_ais_req describing:

```
struct kvm_s390_ais_req {
        __u8 isc;
        __u16 mode;
};
```

isc contains the target I/O interruption subclass, mode the target adapter-interruption-suppression mode. The following modes are currently supported:

- KVM_S390_AIS_MODE_ALL: ALL-Interruptions Mode, i.e. airq injection is always allowed;
- KVM_S390_AIS_MODE_SINGLE: SINGLE-Interruption Mode, i.e. airq injection is only allowed once and the following adapter interrupts will be suppressed until the mode is set again to ALL-Interruptions or SINGLE-Interruption mode.

KVM_DEV_FLIC_AIRQ_INJECT

Inject adapter interrupts on a specified adapter. attr->attr contains the unique id for the adapter, which allows for adapter-specific checks and actions. For adapters subject to AIS, handle the airq injection suppression for an isc according to the adapter-interruption-suppression mode on condition that the AIS capability is enabled.

KVM_DEV_FLIC_AISM_ALL

Gets or sets the adapter-interruption-suppression mode for all ISCs. Takes a kvm_s390_ais_all describing:

```
struct kvm_s390_ais_all {
       __u8 simm; /* Single-Interruption-Mode mask */
       __u8 nimm; /* No-Interruption-Mode mask *
};
```

simm contains Single-Interruption-Mode mask for all ISCs, nimm contains No-Interruption-Mode mask for all ISCs. Each bit in simm and nimm corresponds to an ISC (MSB0 bit 0 to ISC 0 and so on). The combination of simm bit and nimm bit presents AIS mode for a ISC.

KVM_DEV_FLIC_AISM_ALL is indicated by KVM_CAP_S390_AIS_MIGRATION.

Note: The KVM_SET_DEVICE_ATTR/KVM_GET_DEVICE_ATTR device ioctls executed on FLIC with an unknown group or attribute gives the error code EINVAL (instead of ENXIO, as specified in the API documentation). It is not possible to conclude that a FLIC operation is unavailable based on the error code resulting from a usage attempt.

Note

The KVM_DEV_FLIC_CLEAR_IO_IRQ ioctl will return EINVAL in case a zero schid is specified.