

APÉNDICE: FUNCIONES DE OPNET

A continuación presentamos algunas instrucciones e interrupciones propias de OPNET que pueden sernos útiles para entender el diseño de los procesos que conforman los distintos nodos implementados en algunas de nuestras prácticas.

- ❑ **INSTRM**: hace referencia a una cadena de entrada. Utilizaremos 0 para la cadena de entrada de paquetes de datos y 1 para la de ACKs.
- ❑ **OUTSTRM**: se utiliza para referirnos a las cadenas de salida
- ❑ **#define NOMBRE_DE_LA_INTERRUPCION (definición)**: establece en qué condiciones (*definición*) se produce la interrupción denominada *NOMBRE_DE_LA_INTERRUPCION*.
- ❑ **Op_intrpt_type()**: sirve para determinar de qué tipo es la interrupción.
- ❑ **OPC_INTRPT_STRM**: se emplea para indicar que la interrupción se produce como consecuencia de un evento en una cadena, sea de entrada o de salida.
- ❑ **Op_intrpt_strm()**: permitirá indicar en cuál de las cadenas ha de suceder el evento para que se produzca la interrupción.
- ❑ **OPC_INTRPT_SELF**: la interrupción es del tipo propio, esto es, ha sido planificada por nosotros mismos en algún punto del código C de nuestro proceso.
- ❑ **Op_intrpt_code()**: será un identificador que permita distinguir entre las distintas interrupciones planificadas por nosotros mismos, siempre que haya varias.
- ❑ **OPC_INTRPT_ENDSIM**: la interrupción se producirá cuando se alcance el fin de la simulación.
- ❑ **Op_strm_empty(nombre_cadena)**: será una interrupción que se activará en el momento en que la cadena "nombre_cadena" se encuentre vacía.
- ❑ **Int, double, Packet *, Objid**: se utilizan para definir las distintas variables de las que haremos uso en nuestro programa, indicando el tipo de variable que constituyen, entero, doble, puntero a paquete o identificador de objeto, son las que aparecen en nuestros procesos.
- ❑ **Extern**: al preceder a uno de los términos anteriores hace que sea posible que dicha variable sea accesible desde otro proceso, pudiendo conocer su valor en todo momento.
- ❑ **Void**: lo emplearemos en el *Function Block* con el fin de definir las funciones.
- ❑ **Op_stat_scalar_write("nombre",variable)**: Permite guardar con el nombre "nombre", en el fichero Scalar File que designemos en la simulación, los valores que toma la variable tras la simulación.
- ❑ **Op_pk_get(cadena)**: Obtiene, lee, un paquete de la cadena, devolviendo un puntero a dicho paquete.

- ❑ ***Op_pk_nfd_set(puntero_paquete,"nombre_campo",valor)***: Hace que el paquete al que apunta el puntero_paquete tome para el campo "nombre_campo" un valor igual a "valor".
- ❑ ***Op_pk_nfd_get(puntero_paquete,"nombre_campo",&variable)***: Guarda en la variable el contenido del campo "nombre_campo" del paquete al que apunta el puntero_paquete.
- ❑ ***Op_pk_total_size_set(puntero_paquete,tamaño)***: Asigna al paquete apuntado por el puntero_paquete una longitud en bits igual a tamaño.
- ❑ ***Op_pk_create_fmt("tipo_paquete")***: Crea un puntero a un paquete, cuyo formato es "tipo_paquete".
- ❑ ***Op_pk_send("puntero_paquete",cadena)***: envía el paquete al que se refiere el puntero_paquete a través de la cadena.
- ❑ ***Op_pk_destroy(puntero_paquete)***: destruye el paquete.
- ❑ ***Op_sim_time()***: devuelve el instante de tiempo de la simulación.
- ❑ ***Op_pk_creation_time_get(puntero_paquete)***: devuelve el instante de tiempo en que se
- ❑ creó el paquete.
- ❑ ***Op_intrpt_schedule_self(instante_de_tiempo,código_identificativo)***: Planificamos una interrupción que se producirá en el instante de tiempo indicado y que queda identificada por el código identificativo).
- ❑ ***Op_pk_copy(puntero_paquete)***: Copia un paquete en otro.
- ❑ ***Op_intrpt_clear_self***: Desactiva todas las interrupciones propias planificadas.

Si aparece alguna instrucción de la que no se conozca el significado, se habrá de acudir al manual de *OPNET > Product Documentation > Search*.