

Objective 1 Directions

Add_event

```
void Add_event(struct simtime *time, int event, int agent )
```

This function inserts a future event in the list new_events in the proper time sequence. new_events points to the end of the list having the smallest time defined by the given function:

```
    Cmpr_time(struct simtime * , struct simtime *)
```

Directions:

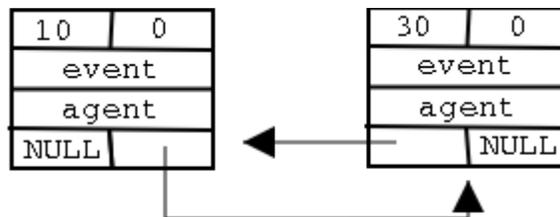
1. This function is called by Load_events(void)
2. Use the structure event_type with the given simtime, agent, and event.

```
/* The event list is a doubly-linked list of elements of EVENT_TYPE */
struct event_type {
    struct simtime    time;
    int               event;
    int               agent;
    struct event_type *prev,*next;
};
```

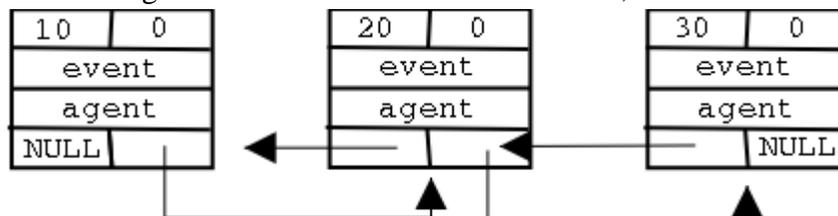
3. Insert it at the appropriate position in the event list (new_events). The event list is ordered chronologically so make sure to maintain the correct order while inserting by using the provided function:

```
    cmpr_time(struct simtime, struct simtime)
```

Before:



After inserting a simtime record with seconds = 20, nanosec = 0



Load_events

```
void Load_events(void)
```

This function is called from simulator.c (The simulator driver) and it initializes the event list (new_events) from the file logon.dat. This file normally contains only LOGON events for all terminals. However, for debugging purposes, logon.dat can contain events of any type. This function uses:

```
Add_event(struct simtime * , int, int)
```

Directions:

- Refer to intro.doc for the logon.dat format
 - Use the given function:

```
convrt_time(struct simtime * time1, long time2)
```
 - The event name and agent name can be either in upper or lower case or a combination. Make sure you convert it to upper case.
1. Convert the event name to eventid using the eventidtab[] defined in simulator.c. Example: event name = LOGON, event id = 0
 2. Convert the agent name to agent. Here two cases arise:
 1. If the agent name is Uxxx, agent id = xxx. (agent is a user)
 2. If the agent is a device, then: TRMSIZE + 1 < agent < TRMSIZE + DEVSIZe where TRMSIZE is the number of terminals (users) and DEVSIZe is the number of devices. You will have to use the lookup table devtable defined in simulator.c.
 3. Call Add_event(time2, enevt_id, agent_id) to build the event list.

Write_event

```
void Write_event(int event, int agent, struct simtime *time)
```

This function writes an event to "simout" with the format:

```
"EVENT AGENT TIME (HR:xxxxxxxx MN:xx SC:xx MS:xxx mS:xxx NS:xxx"
```

You will have to convert the nanosec field to MS, mS, and NS. The seconds field will have to be converted to HR, MN, and SC.

Directions:

1. Called from Interrupt(void)
2. Convert the event_id and agent_id to event name and agent name for printing to the output file simout which is already open.

Interrupt

```
void Interrupt(void)
```

This function is called from simulator.c (The simulator driver)

This function:

- 1. removes an event from new_events*
- 2. sets CLOCK, AGENT, and EVENT*
- 3. deallocates the event element*
- 4. writes the event to "simout"*
- 5. Copies CPU.mode and CPU.pc into oldstate*
- 6. Copies newstate into CPU.mode and CPU.pc*

Directions:

1. This function is called from simulator.c