Processes in Linux

The parent process is the init process which is started and whose ID is 1
All the processes are identified by their ID's called PID's.
Many information is required to describe a process and for that a data structure is used called process_descriptor and in the linux code it is defined as task_struct.

Process List : All the existing processes are contained in the process list which is a double linked list. And it is circular and the first element is the init process and the last is the recently inserted element.

How the process are run?

Initially they made run queues which contained all the processes in the running state and the scheduler searched through the list and found the best runnable process. This was feasible but inefficient so to make the operation O(1) they made some changes.

As the priority of the process is an important factor. They made queues or list according to the priorities which means that 140 lists were made and processes of the same priority were inserted in the same list.

How kernel identifies the current running process?
There is a current variable and the current running process data is stored in that variable.

PID Hash Tables : To access the process descriptor via the PID value instead of searching through the whole process list. PID hash tables are defined. These hash tables are of 4 types

PID : Process ID
TGID : PID of the Thread Group leader
PGID : PID of the Process group leader.
SID : PID of the session group leader.

They are useful if you want to find all the processes linked to a particular login session or a particular thread group.

The hashing is performed by the pid_hashfn which uses the hash_long function and the associated file is They use the hash long function which is multiplying the pID by a very large Prime number and extracting the high bits of that number to generate a hash value.

Linux uses hashing with chaining to avoid collision so basically each entry is a double linked list pointer.

These are the four hash tables defined and their addresses are stored in the pid_hash variable.

Check the PID hash structure.

How Linux runs the process scheduler and runqueue
Runqueue is a queue which contains a number of lists each list corresponding to a priority so there are 140 lists from 0 to 139 and a process with priority 1 will go to list 1 and vice versa.

Wait Queues : The process waiting for an event to continue are put in the wait queues. Events may be an interrupt or synchronization or some system resources to be released. The wait queues are implemented as double linked list.
The wait queues implemented in the linux kernel code is as follows :

```
/*
 * A single wait-queue entry structure:
 */
struct wait_queue_entry {
        unsigned int            flags;
        void                    *private;
        wait_queue_func_t       func;
        struct list_head        entry;
};

struct wait_queue_head {
        spinlock_t              lock;
        struct list_head        head;
};
typedef struct wait_queue_head wait_queue_head_t;
```

So the single wait_queue_entry is defined which contains the following fields

1. *private which will contains the address of the process descriptor.
2. Func will describe the event for which the process is waiting and when to wake up and for most process default wake_up_function is assigned.
3. Flags determine whether it is an exclusive process or non exclusive process.

Why is wait_queue_head defined separately?

Types of sleeping processes:

The processes are waiting for resources or some event and there can be many processes waiting for the same event so it is not convenient to wake up all the sleeping processes when the resource becomes available because if two processes are waiting for the exclusive access to the resource then the second process has to go to sleep again.

Exclusive processes: The