

A. Basic functionality and general questions

1. What is the main advantage of moving the scheduler out of kernel-mode?
  - The main advantage is that the scheduler can be customized to optimize for the kind of work to be done by the OS.
2. What is the main disadvantage of moving the scheduler to user-mode?
  - The main disadvantage is that it is a security risk to allow the users to intervene in the scheduling of processes.
3. Explain what is the operation performed when SCHEDULING\_START is invoked. (check schedule.c in the same folder)
  - SCHEDULING\_START: is a special case for the system processes, in which the quantum and priority are set explicitly rather than inherited from the parent.
4. What is the default quantum assigned to user-level processes?
  - The default is 200 microseconds defined by:  
`#define DEFAULT_USER_TIME_SLICE 200`
5. What is the priority assigned to user-level processes? Others?
  - Priority assigned to user level:
    - a. SCHEDULE\_CHANGE\_PRIO 0x1
  - Others:
    - a. SCHEDULE\_CHANGE\_QUANTUM
    - b. SCHEDULE\_CHANGE\_CPU
  - If the method receives a process with priority greater than the priority of the father it will schedule the new process with priority greater than the father, else it will schedule the father.
6. What is the function in charge of scheduling processes? (invoked at do\_start\_scheduling)
  - The function in charge of scheduling processes is called `schedule_process(struct schedproc * rmp, unsigned glags)`

7. Identify system files. Describe the main objective of each, and provide details of how it is implemented.

- /usr/lib/src/libsys/sys\_schedule.c
  - a. The function will receive the endpoint\_t proc\_ep, the priority, quantum and cpu. It will use the functions SCHEDULING\_ENDPOINT, SCHEDULING\_PRIORITY, SCHEDULING\_QUANTUM AND SCHEDULING\_CPU functions using the data received to return the system scheduled with “return(\_kerel\_call(SYS\_SCHEDULE< &m));”.

- /usr/src/kernel/system/do\_schedule.c

Describe the main objective of each, and provide details of how it is implemented.

- a. It uses the scheduler process which is the only process who can schedule it. To schedule the process it use the scheduling\_priority, scheduling\_quantum, scheduling\_cpu and it will return “sched\_proc(process,priority, quantum, cpu)”.

8. Identify function sched\_proc (/usr/kernel/system.c) and explain its purpose.

- The function sched\_proc start checking the parameters priority, quantum and CPU to be valid. After checking the parameters, the function verifies if the process is runnable and it assigns a new priority and quantum.

## B. Advanced Features

1. What is the policy in MINIX to manage those processes that have exhausted all of their quantum (out-of-quantum)?

(follow the invocation of SCHEDULING\_NO\_QUANTUM in main.c)

- When the process finds itself in this state, the do\_noquantum will add a level of priority to the process in the queues to force the processes of lesser priorities to run previous to his processes reactivation.

2. Explain the purpose of the scheduling queues. (see Fig. 2)

- This function is called every N ticks to rebalance the queues. The current
  - \* scheduler bumps processes down one priority whenever they run out of
  - \* quantum. This function will find all processes that have been bumped down,
  - \* and pulls them back up. This default policy will soon be changed.

3. Describe the steps that the scheduler takes to increase the priority of a process. (see balance\_queues in schedule.c)

- This function is called every 100 ticks to rebalance the queues. The current scheduler bumps processes down one priority whenever they run out of quantum. This function will find all processes that have been bumped down, and pulls them back up.
4. Describe the functionality of function `pick_proc` (see `/usr/src/kernel/proc.c`)
- Decide who to run now. A new process is selected and returned when a billable process is selected, record it in `'bill_ptr'`, so that the clock task can tell whom to bill for system time. This function always uses the run queues of the local CPU.