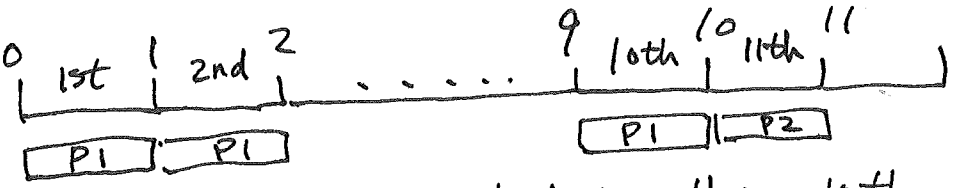
Lecture 34 project 5-2

1. time slot vs. time instant

<time 10> process 1 is finished...
<time 10> process 2 is running



process 1 is finished in the 10th time slot.
2 is running " " 11th time slot.

To use "time instant" rather than "time slot",
If you use "time slot", ~~it~~ the output should be.
<time slot 10> P1 is finished.
<time slot 11> P2 is running.

2. show sample code input.c

gcc -g input.c -o input
./input task.list
./input task1.list
cat task.list

2.1 argv[1] : file name
2.2 fopen → fp : FILE
2.3 input : fscanf(fp, "%u", &task_array[0].pid) == EOF
unsigned int → u_int
task{ pid, arrival_time, burst_time } ← task_array[32] : task
"press ~~key~~ any key to continue..." getchar().
2.4
2.5 fclose(fp).
2.6 goto line 62 remove & demo "core dumped" (segmentation fault)

3 Naming : scheduling policy functions.
don't use uppercases. (constant only).
int FCFS( ); fcfs-policy( ); ~~Srtf~~-policy( );
Srtf

Lecture 34 project 5-2

data structure { task list : task_array
task : task_t : task_info struct

2: task_info_loader(file_name, task_array[], count);
see input.c (input) (* output) (input)
finish_array[] count

4: fcfs_policy(task_array[], ~~task~~ task , count, quantum);

5: srtf_policy(_____ , _____ , _____ , _____);

6: ~~roundrobin~~ rr_policy (task_array, finish_array, count, quantum);

7 Compute_stat_info (finish_array, count, * stat_info);

8 display_stat_info (stat_info);

D6 stat_info_t {
avg_waiting_time : u_int
avg_response_time : u_int
avg_turnaround_time : u_int
cpu_usage : u_int

D2 policy_t : enum
~~enum policy~~ typedef enum {
FCFS,
SRTF,
RR
} policy_t

Simulator : coordinator (task_array, count, quantum, finish_array);
↳ compute_stat_info() display_stat_info() are outside of the coordinator).
/* why no finish_array. */
Coordinator (task_array, count, quantum);
↓ calls {
fcfs_policy();
srtf_policy();
rr_policy();
compute_stat_info();
display_stat_info();