OS Godz – R2 – Programmer's Manual

Nathan Fielding – Logan Yokum – Dylan Smith – Ethan Boddy

---

void pcb_op(char* pcb_str)

Function to manage user commands associated with process control blocks. Takes in a character pointer as the command and returns nothing. The function compares the parameter string with each of the user process commands and determines which command to execute. If an invalid command is input then the user is prompted to enter a valid command.

void pcb_create(const char* name, int class, int priority)

Function to create a new process. Takes in a constant character pointer as the name of the process, an integer to designate its class, and an integer to set its priority. The function returns nothing. The process name is used to verify that a process with the same name does not already exist, integer class parameter is validated to be between 0 and 1, and the integer priority parameter is validated to be between 0 and 9. The new process is then created by calling pcb_setup and inserted into the appropriate queue.

void pcb_delete(const char* name)

Function to remove a process from the queue. Takes in a constant character pointer as the name of the process and returns nothing. The process name is used to find the process, check the suspension status, and finally remove it.

void pcb_block(const char* name)

Function to change the state of a process to blocked. Takes in a constant character pointer as the name of the process and returns nothing. The process name is used to find the process, verify that the status is not already blocked, remove it from the ready queue, and insert into the blocked queue.

void pcb_unblock(const char* name)

Function to change the state of a process to not blocked. Takes in a constant character pointer as the name of the process and returns nothing. The process name is used to find the process, verify that the status is not already unblocked, remove it from the blocked queue, and insert into the unblocked queue.

void pcb_suspend(const char* name)

Function to change the state of a process to suspended. Takes in a constant character pointer as the name of the process and returns nothing. The process name is used to find the process, verify that the status is not already suspended, and insert into the suspended queue.

void pcb_resume(const char* name)

Function to ensure that a process is ready and not blocked or suspended. Takes in a constant character pointer as the name of the process and returns nothing. The process name is used to find the process, verify that the class status is not already ready, and insert into the ready queue.

void pcb_set_priority(const char* name, int priority)

Function to set the priority value of a process. Takes in a constant character pointer as the name of the process and an integer as the priority value to be set. The function returns nothing. The process name is used to find the process and the integer parameter is validated to be between 0 and 9. The new priority member is then set and the process is inserted into the appropriate queue.

void pcb_show_pcb(const char* name)
Function to display all of the members of a given process. Takes in a constant character pointer as the name of the process and returns nothing. The process name is used to find the process and access all of its members. sys_rec is then used to display each of the members of the process (name, class, priority, state).

struct pcb* pcb allocate()
Function to allocate memory for a process. Takes in no parameters and returns the struct for which the memory was allocated. sys_alloc_mem is called to allocate memory.

int pcb free(struct pcb* p)
Function to free a process from memory. Takes in a process as a struct pointer and returns an integer flag value to confirm that the memory was freed. sys_free_mem is called to free memory.

struct pcb* pcb setup(const char* name, int class, int priority)
Function to set up a struct with the appropriate members. Takes in a constant character pointer as the name, an integer to designate the class, and an integer to set the priority and returns the process as a struct. Memory is allocated for the process and each member (class, priority, state) is assigned.

struct pcb* pcb find(const char* name)
Function to find a process in the system. Takes in a constant character pointer as the name of the process and returns the process as a struct. list_find is called to check the head node for each possible state (ready, blocked, suspended/ready, suspended/blocked).

void pcb insert(struct pcb* p)
Function to insert a process into the appropriate queue. Takes in a process as a struct pointer and returns nothing. Checks for the state of the process and calls list_insert to insert the process into the appropriate queue.

int pcb remove(struct pcb* p)
Function to remove a process from a queue. Takes in a process as a struct pointer and returns an integer flag to confirm that the removal was successful. Checks each state of the head process in the appropriate queue and calls list_remove to remove a node from the queue.

int list_free(struct pcb* head)
Function to free a process from memory. Takes in a process as a struct pointer and returns an integer to confirm that the memory was freed successfully. The function starts at the head node and checks each process until the appropriate node is found and the memory is freed.

pcb *list_find(struct pcb* head, const char* name)
Function to find a process in a queue. Takes in a struct pointer as the head node and a constant character pointer as the name of the process. The function returns the process as a struct pointer. The function iterates through a queue to find the parameter process.

void list_insert(struct pcb** head, struct pcb* p)
Function to insert a process into a queue. Takes in a struct pointer to a pointer as the head node and struct pointer as the process. The function returns nothing. The queue is iterated through until the appropriate position is reached for the process to be inserted.

int list_remove(struct pcb** head, struct pcb* p)
Function to remove a process from the queue. Takes in a struct pointer to a pointer as the head node and a struct pointer as the process. The function returns an integer flag value to confirm that the removal was successful. The queue is iterated through until the appropriate position is reached for the process to be removed.

void pcb_show_ready()
Function to show the processes in a ready state. Takes in no parameters and returns nothing. The function calls sys_rec to display all of the processes associated with a ready state.

void pcb_show_blocked()
Function to show the processes in a blocked state. Takes in no parameters and returns nothing. The function calls sys_rec to display all of the processes associated with a blocked state.

void pcb_show_all()
Function to display all of the processes in the system. Takes in no parameters and returns nothing. The function calls sys_rec to iterate through and display all of the processes in the system.