CS450 Group 1

Matt Angeline
Kyle Breedlove
Katrina Cortopassi
Sampurna Dhungana

Programmer's Manual

Contents

Serial.c . . . . . Page 2
Keyboard_polling.c . . . . . Page 2
Comhand.c . . . . . Page 3
ProcessQueues.c . . . . . Page 4
UserCommands.c . . .. Page 5

Serial.c

*int polling(char * buffer, int * count):* This function calls the polling function from the Keyboard_polling.c file written as part of R1.

Keyboard_polling.c

*int keyboard_polling(char * buffer, int * count):*This function that manages the input and output of the OS by reading and echoing the user's input in the terminal through the serial port.

Comhandle.c

int comHand(): This function is used to display the menu to the user, then asks the user for his input using sys_req(Read…). Once the input is entered from polling(), it will then check the input at the first location inside of the userInput array and determine if it equals one of the options available. Each option is labeled within the menu for the user to choose from. After each execution except for shutdown will automatically return to comhandle to allow the user to make another selection. Shutdown asks the user if they want to shut down then either enters the shutdown protocol or returns to the comhandler.

void help(): Writes each user function to screen along with an explanation on what they do using sys_req(WRITE,...).

void clearInput(): Clears the user input from the buffer.

void getTime(): This function writes the current value of time to the RTC index register, converts the result to decimal value and displays the decimal value of time to the user.

void setTime(): Prompts the user to enter a time which is then stored in memory.

*void getDate():*This function writes the current value of date to the RTC index register, converts the result to decimal value and displays the decimal value of date to the user.

void setDate(): Prompts the user to enter a date which is then stored in memory.

void version(): Displays current version of the operating system using sys_req(WRITE,...).

void shutdown(): Begins the shutdown process, by prompting the user to confirm that they want to shutdown the system using sys_req(WRITE,...).

unsigned int decToBCD(int num): converts a given integer (num) to binary coded decimal. Returns an unsigned integer.

int BCDToDec(int num): converts a given integer to its corresponding character. Returns a character.

void itoa(int num, char str):* Takes in an integer number and a character pointer as parameters. Then it gets the second number using modulo 10 and the first number by dividing by 10. Finally, it stores the characters in the character array.

processQueues.c

Struct pcb: This struct defines a PCB: its name, class, priority, status, state and stack.

Struct queue: This struct defines a queue: its head, tail and count.

queue getReadyQueue():* This function returns a pointer to the ready queue.

queue getBlockedQueue():* This function returns a pointer to the blocked queue.

pcb allocatePCB():* This function is used to allocate the required memory for the PCB.

int freePCB(pcb PCB):* This function is used to free the memory allocated to the PCB.

pcb setupPCB(char name[],int pcbClass,int Priority):* This function is used to allocate memory to the PCB and assign its class, priority state and status.

pcb findPCB(char[] name):* This function searches through the ready and blocked queues to find the PCB with a matching name to the char[] passed in as a parameter. It will return a pointer to this PCB or NULL if it cannot be found.

void addToReadyQueue(pcb PCB):* This function adds the PCB passed in as a parameter to the ready queue sorting by its priority compared to the priorities of the other PCBs in the queue.

void addToBlockedQueue(pcb PCB):* This function adds the PCB passed in as a parameter to the blocked queue following First In First Out logic.

int removeFromQueue(pcb PCB):* This function determines which queue the PCB is in by looking at its state. It then will set the surrounding PCBs' previous and next PCB pointers to their new previous and next PCB pointers, removing it from the queue. Returns 0 on error and 1 on success.

userCommands.c

void createPCB(char [] name, int pcbClass,int priority): This function takes in the given parameters and calls setupPCB and adds it to the ready queue.

void deletePCB(char []name): This function locates the PCB within the given queue and removes it from the queue while also freeing its allocated memory.

void block(char [] name): This function locates the given pcb and sets its state to blocked, removes it from its current queue and then reinserts it into the correct queue.

void unblock(char [] name): This function locates the given pcb within the blocked queue and sets its state value to ready, then removes it from the current queue and reinserts it back into the correct queue.

void showPCB(char [] processName): This function checks to ensure that the name of the PCB is a valid PCB within the queues. It then copies each parameter within the PCB requirements and prints them to the screen.

void showReady(): This function locates the head of the ready queue, and then passes in the name of each of the members of the ready queue to the showPCB function.

void showBlocked(): This function locates the head of the blocked queue, and then passes in the name of each of the members of the block queue to the showPCB function.

void showAll(): This function calls showReady() then showBlocked() and writes all pcbs to the screen in the order that they occur within their queues. It displays the pcbs in the ready queue first then the blocked queue.

void Suspend(char [] name): This function locates the given pcb and changes its status to “Suspended”.

void Resume(char [] name): This function locates the given pcb and changes its status to “Not Suspended”.

void setPriority(char name[], int PCBpriority): This function locates the given pcb and changes the priority number to the new given number.