OPERATING SYSTEMS - CS3193
REPORT

N JAWAHAR SAI
P SAGAR REDDY
K LIKITH KUMAR
T ABHILASH

INTRODUCTION

In this OS Lab course, we have improved the XINU OS to have extra functionalities like a dummy file system, basic terminal commands like ls, cd, pwd etc.. and a search algorithm that searches for files and folders present in the current directory.

FEATURES:

1. CD :
    a. *CD* (change directory) feature allows the user to change the directory either to any of the directories present in the current directory or to the previous directory.
    b. Usage :
        i. cd <*directory name*> - to move to a specific directory.
        ii. cd .. - to move to the previous directory.
        iii. Ex: "*cd Downloads*", "*cd ..*"
2. LS :
    a. *LS* feature allows the user to list all the files and directories present in the current directory.
3. PWD :
    a. *PWD* feature is used to find the path of the current directory.
4. MKDIR :
    a. *MKDIR* feature is used to create a new directory in the current directory.
    b. Usage :
        i. mkdir <*directory name*>
        ii. Ex: *"mkdir Desktop", "mkdir new_dir"*
5. CREATE :
    a. *CREATE* feature is used to create a new file in the current directory.
    b. Usage :
        i. create <*file name*>
        ii. Ex: *"create xinu report", "create slides"*
6. SEARCH :
    a. *SEARCH* feature is used to search for files and folders present in the current directory.
    b. Usage:
        i. search <*search query*>
        ii. Ex: *"search report", "search xinu report"*

CONTRIBUTION

- Jawahar sai -
  - Search, cd functionality
- Sagar reddy -
  - memory allocation and access
- Likith kumar raju -
  - pwd, create functionalities
- Abhilash -
  - ls, mkdir functionalities

REVISED TOP DOWN

S/no	XINU Features	Related files	Justification
1.	cd	xsh_changedir.c	This feature is used to change current directory.
2.	ls	xsh_listdir.c	This feature is used to list all the files and folders present in the current directory.
3.	pwd	xsh_pathfinder.c	This feature is used to find the path of the current directory.
4.	mkdir	xsh_mkdir.c	This feature is used to create a new directory in the current directory.
5.	create	xsh_createfile.c, mkbuffpool.c , getbuf.c	This feature is used to create a new file in the current directory.
6.	search	xsh_search.c	This feature is used to search for files and folders present in the current directory.

REVISED BOTTOM UP

S/no	OS Features	Files used	Justification
1.	Process Scheduling	resched.c, queue.c	Used to schedule all the running processes based on priority of each process.
2.	Context Switching	ctxsw.S	Used when switching processes to store and access the states of swapped processes.
3.	Memory allocation	mkbuffbool.c	Used to create buffers in memory to store the names of files and folders of dummy file system.
4.	Memory access	getbuf.c	Used while allocating a dummy folder for accessing already created buffer space.
5.	I/O system	getc.c, control.c, read.c, write.c, open.c, close.c	Used to take input from the keyboard and to display files or applications matching the search query and other terminal functionality outputs on the screen.
6.	Process synchronization	wait.c , signal.c	Used to synchronize main process and child processes such that the main process executes first and allows all the child processes to run one after another in the sequence assigned by the main process.
7.	Inter process communication	send.c , receive.c , recvclr.c	Used to pass messages from child process to main process, to let the main process know if the child process has found any matches for the search query.

IMPLEMENTATION

S.No	Project features description	Xinu files modified	Xinu files used	New files added
1.	search	shprototypes.h , shell.c , process.h , initialize.c	send.c , receive.c , recvclr.c , resched.c , ctxsw.S , process.h , initialize.c, semcreate.c, kprint.c , wait.c , signal.c	xsh_search.c
2.	create	shell.c , shprototypes.h , process.h , initialize.c	main.c , resched.c , ctxsw.S , process.h , initialize.c , mkbuffpool.c , getbuf.c , kprintf.c	xsh_createfile.c
3.	ls	shprototypes.h , shell.c	resched.c , ctxsw.S , process.h , initialize.c , kprintf.c	xsh_listdir.c
4.	pwd	process.h , initialize.c , shell.c , shprototypes.h	resched.c , ctxsw.S , process.h , initialize.c , kprintf.c	xsh_pathfinder.c
5.	mkdir	shprototypes.h , shell.c	main.c, resched.c , ctxsw.S , process.h , initialize.c , kprintf.c	xsh_mkdir.c
6.	cd	shprototypes.h , shell.c	resched.c , ctxsw.S , process.h , initialize.c , kprintf.c	xsh_changedir.c

Deadlock -

- Necessary conditions for deadlock :
    - Cyclic dependency graph
        - A set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.
- Proof for Deadlock Free Functionality :
    - Resources used - input, output, memory(buff pool)
    - Search functionality :
        - In search functionality, initially only the main process is executed which uses keyboard and memory resources which are not blocked by any other

    process. And after every child process completes the search process it calls for the main process which uses the output resource to display the search result. As no process is being halted because of resources, no cycle is formed. Hence deadlock does not occur for search functionality.

  - In all other functionalities, only one process is executed. As there is only one process, cycle is not formed. Hence deadlock does not occur.
  - Hence deadlock does not occur in any functionality.