Hacettepe University
Computer Science and Engineering Department

Name and Surname	:Serdar Gül
Identity Number	:20421689
Course	:Bil-235
Experiment	:3. experiment
Date	:14.11.2007
Advisors	:Prof. Dr. Ali SAATÇİ,Erkan YASAN
Main Program	:search.c tokenize.c. and traverse
Subject	:Operating Systems
Developing Platform	:Eclipse 3.2 IDE,LINUX
Mail Address	:fuchserdar@gmail.com

2. Software Using Documentation

2.1 Software Usage

This program is very useful.This is about operating systems.With using LINUX shell commands and writing some source code with using the C Programming Language we will make a very small operating system.

This program's usage is so basic.At first user runs the tokenizer and traverse and index the files after this runs the search engine and take the results to the output(console).

This is like this;

```
rm tokenize       # remove old tokenize executable
rm search        # remove old search executable
gcc –o tokenize tokenize.c  # compile and generate new executable
gcc –o search search.c    # compile and generate new executable
traverse tokenize <DIRECTORY> # generate index
search [-i] <token [ (AND | OR) token ] >
search [-i] <token [ (AND | OR) token ] >
…
```

2 Error Messages

"there is not any argument in this main program":if user don't enter an argument program gives this message

3.Software Design Notes

3.1.Description of program

3.1.1.Problem

In this experiment our main problem is combining the shell programming in LINUX and the normal programming in ANSI C.

At first we will take a directory and we will take this path.After these operations we will index the files what are in the directory and we will apply this to the all directories who the user wants it.

After this we will make them ordered and we will use their path to the main directory.

And finally we will search the key words which we want it and program gives us a result to the console that if the directory has a file named which we searched like.

3.2.System Chart

INPUT

PROGRAMS

OUTPUT

<input device> keyboard

<program names>search.c tokenize.c traverse

<output devices>monitor and indexedFile.txt

3.3.Main Data Structures

In this problem I use

A stringTokenizer that is a struct that has some functions that is used to tokenize a string given to it.

And also that I use some functions and I define it in the header file.

And I use some of the

These are the functions which are used to cope with the problem given to us

Search():this function searchs some data which is given by the user and searchs in the file names which are indexed in the directory and gives the results to the console that does any file have it.If any file has this word or words write their paths to the console.

Tokenize():this function tokenizes the whole words of the filename and designs to a path like this

C:\serdar\uml.pdf – this is the path

And w tokenize it all the things between any sign like (, = } and etc.

3.4 Algorithm

Program algorithm can be like this.

1.at first go to tokenize function

1.1.take the argument
1.2.check the reliability of the argument
1.2.1. if it’s right go to 1.3. step
1.2.2. else go to 1.1. step
1.3. take whole string and go to 1.4. step
1.4. take a char and control if it is a letter or a sign
1.4.1.if letter go to 1.4. step
1.4.2.else go to 1.5. step
1.5. token this part of the string
1.6. go to 1.1. step until whole directory will be finished

2. go to the traverse part
2.1. this is used to make tokenizing.

3. go to the search part
3.1. take the argument
3.1.1.if this argument is just “search”
3.1.1.1.go to the indexed file and
3.1.1.2.if there is a file contains the parameter
3.1.1.2.1.write whole path to the console
3.1.1.3.else
3.1.1.3.1.write nothing
3.1.2.if this argument is “search -i”
3.1.2.1. control that argument comtains “AND” or “OR”
3.1.2.1.1.if has nothing
3.1.2.1.1.1. go to the indexed file and
3.1.2.1.1.2.if there is a file contains the parameter uppercase or lowercase
3.1.2.1.1.3. write whole path to the console
3.1.2.1.1.4else
3.1.2.1.1.4.1.write nothing
3.1.3.1.1.if has “AND”
3.1.3.1.1.1. go to the indexed file and

3.1.3.1.1.2.if there is a file contains the both two parameters uppercase or lowercase
3.1.3.1.1.3. write whole path to the console
3.1.3.1.1.4else
3.1.3.1.1.4.1.write nothing
3.1.4.1.1.if has “OR”
3.1.4.1.1.1. go to the indexed file and
3.1.4.1.1.2.if there is a file contains any of two parameters uppercase or lowercase
3.1.4.1.1.3. write whole path to the console
3.1.4.1.1.4else
3.1.4.1.1.4.1.write nothing

3.1.3. if argument is “search” with “OR” or “AND”
3.1.3.1.go to 3.1.2. stpe but do the processes without uppercase or lowercase.Find the results however searched it.

4.1. Software Testing Notes

4.1.1 Bugs and Reliability

I can’t do some of the things you wanted for the program.So in some search criteria’s program couldn’t run exactly.

4.1.2 Software extendbility and upgradbility

We can use this for the other languages to change it and use it.In this we can make vectors may be we can change it to the other data structures with a little changes in the code optimizations.

We can make lots of things in any other language except C like C++ or JAVA

We can make this a Graphic User Interface.And the program can be seen pretty.

4.1.3 Performance considarations

It is very fast and we use a lot of useful things like shell codes.

4.1.4 Comments

I think this experiment is very useful for us to develop ourselves.But
I thought that the time given us to do this is so less so I can’t complete the whole program

4.1.5.Some Terms in the Context Of the UNIX

Built in Shell Commands : These commands are used that user can communicate with the UNIX operating system.These are the shell commands.We can make some difference in Shell Layer of UNIX with this commands.These are platfrom independent.we can give an example of GREP command in UNIX

For example;

alias,bind,builtin,command,declare,echo,enable,help,let,local
,logout,printf,read,shopt,type,typeset,ulimit and unalias .

Application : Application is the final form of a source code after compiling and linking.Application is the executable part of the program.After linking the compiled code,we will have an .exe file.If we run this we can say that we are using application.Program setups in UNIX can be given as an example in this subject.For example

Mozilla : internet explorer

Star office : an office program works with;Documents,Spreadsheets, Presentations, HTML and Drawings

Pidgin : Instant Messenger

AbiWord : Word Processor

pico : mail composer editor

vi : a screen-based text editor

ssh : secure shell,a network protocol that allows data to be exchanged over a secure channel between two computers.

Openoffice : UNIX form of the Microsoft Office but it is more useful than Office in my opinion

System Call : Some applications send system calls the operating system to want a resource or a job.And operating system stops the operating of the current command and starts to help the applcation which sends the system call.In UNIX,system calls are C programming language functions because UNIX is written in C programming language.If we want to giva an example we can say that Fork() and exec()

functions are the system calls of the UNIZ operating system.

Sys_read : it is used to read something from a file

Sys_open : it is used to open a file

Sys_write : it is used to write something to a file

Sys_close : it is used to close a file

Sys_mkdir : it is used to make a directory

Sys_rmdir : it is used to remove a directory

Sys_gettimeofday : it is used to learn the system time

Sys_settimeofday : it is used to set the time of a day.

exit(): ends a process and returns a value to its parent

Sys_signal(): for sending and receiving software interrupts

kill(): kills the process

Library Functions : UNIX provides a large number of C functions as libraries.These functions are tested so it reduces the debugging time in applications.We can give some examples that

abs():is used to take the absolute value of the number

ctime():is used to take the clock time of the system

stdio():is used to make standart input output operations like;

printf():it is used to print a value to the console

scanf():it is used to take a value from user from keyboard

puts():is used to print a character to console

malloc():is used to take memory from the main memory to used dynamic programming logic

sin():is used to find the sinusodial value of the number we entered

pow():is used to find the power of number we entered

are some of the UNIX Library Functions.

5.References

http://www.codeproject.com/csharp/AlgorithmReuse.asp
http://www.neu.edu.cn/cxsj/materal/otherc/imada/subsection3_15_3.html
http://www.webopedia.com/TERM/A/application.html
http://www.tldp.org/LDP/intro-linux/html/index.html
http://www.linux.org.tr/
linuxcommand.org/learning_the_**shell**.php
linux.about.com/od/**commands**/**Linux_Commands**_and_**Shell_Commands**.htm
www.softpanorama.org/Internals/unix_**system_calls**.shtml
docs.cs.up.ac.za/programming/asm/derick_tut/sys**calls**.html

These are the web sites I visited.

Bilgisayar İşletim Sistemleri by Ali SAATÇi

This is the book I used it.