

# UNIX Shell and History Feature

Source Code |page 2

Wireframes |page 4

## **Submitted by:**

Ailen Grace T. Aspe  
BSEC-4  
2013-1364

## **Submitted to:**

Ms. Margie S. Arda  
CSC 155 Instructor

## SOURCE CODE

```
/*
    Author: Ailen Grace Aspe
    ID: 2013-1364
*/

#include<sys/types.h>
#include<stdio.h>
#include<unistd.h>

#define MAX_LINE 80

char history[10];
char buffer[10];

int main(){

    int should_run =1;
    char line[MAX_LINE];
    char *argv[MAX_LINE]; //pointer array which points to each arguments passed to the program
    int i =0;

    while(should_run)
    {
        printf("osh>");
        gets(line);          //get the string from standard input

        parseCommand(line, argv); //pass the value to a function
        strcpy(history[i], *argv);
        i++;

        if(strcmp(argv[0], "exit") == 0){
            exit(0);
        } //compare the value of argv[0] is equal to exit

        if(strcmp(argv[0], "history")==0){
            printf("***Shell command history***\n");
            int j=1;

            while(i != 0){
                --i;

                printf("%d\t%s\n",j, history[i]);
                j++;
            }
        }

        executeCommand(argv);
    }
}
```

```

    }
}

void executeCommand(char **argv)
{
    pid_t  pid;
    int     status;

    pid =fork();

    if (getpid() < 0) {
        printf("Error");
        exit(1);
    }
    else if (pid == 0) {
        //child process executing
        int val = execvp(*argv, argv);

        if(val<0){
            if(strcmp(*argv[0], "history")){
                printf("Hello");
            }
            else{
                printf("Execution failed\n");
            }
        }
    }
    else {
        //wait for the child process
        wait(NULL);
    }
}

/*
char *line points to the value of command line inputted
**argv points to the value of each argument passed
*/
void parseCommand(char *line, char **argv)
{
    while (*line != '\0') {
        /* Note that \0, terminating null byte*/
        while (*line == ' ' || *line == '\t' || *line == '\n')
            *line++ = '\0';
        *argv++ = line;
        /* save the argument position */
        while (*line != '\0' && *line != ' ' &&
            *line != '\t' && *line != '\n')
            line++;
    }
    *argv = '\0';
    //end of the argument list
}

```

## Wireframe

Execution done in Code blocks

```
osh>cal

      January 2017
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

osh>ls

mp1_aspe  mp1_aspe.c  mp1_aspe.o  mp1_aspe.odt
osh>date

Sat Jan  7 14:22:10 PHT 2017
osh>time

Usage: time [-apwV] [-f format] [-o file] [--append] [--verbose]
          [--portability] [--format=format] [--output=file] [--version]
          [--quiet] [--help] command [arg...]
osh>clock

osh>man fork

osh>man bash

osh>history

***Shell command history***
1      history
2      man
3      man
4      clock
5      time
6      date
7      ls
8      cal
osh>exit

Process returned 0 (0x0)   execution time : 106.423 s
Press ENTER to continue.
```

