Ex.No:3Programs using the following system calls of UNIX operating system fork, exec, getpidAIM: To write C Programs using the following system calls of UNIX operating system fork, exec, getpid. 1. PROGRAM FOR SYSTEM CALLS OF UNIX OPERATING SYSTEMS (OPENDIR, READDIR, CLOSEDIR)ALGORITHM:STEP 1: Start the program.STEP 2: Read the value of n.STEP 3: Calculate ‘r=expr $n%2’.STEP 4: If the value of r equals 0 then print the number is evenSTEP 5: If the value of r not equal to 0 then print the number is odd.PROGRAM:#include<stdio.h>#include<dirent.h>struct dirent *dptr;int main(int argc, char *argv[]){char buff[100];DIR *dirp;printf(“\n\n ENTER DIRECTORY NAME”);scanf(“%s”, buff);if((dirp=opendir(buff))==NULL){printf(“The given directory does not exist”);exit(1);}while(dptr=readdir(dirp)){printf(“%s\n”,dptr->d_name);}closedir(dirp);}SAMPLE OUTPUT:Cc opdir.c./a.outENTER THE DIRECTORY NAME UNITCHAP1.CCHAP2.C2. PROGRAM FOR SYSTEM CALLS OF UNIX OPERATING SYSTEM (fork, getpid, exit)ALGORITHM:STEP 1: Start the program.STEP 2: Read the value of year.STEP 3: Calculate ‘b=expr $y%4’.STEP 4: If the value of b equals 0 then print the year is a leap yearSTEP 5: If the value of r not equal to 0 then print the year is not a leap year.PROGRAM:#include<stdio.h>#include<unistd.h>main(){int pid,pid1,pid2;pid=fork();if(pid==-1){printf(“ERROR IN PROCESS CREATION \n”);exit(1);}if(pid!=0){pid1=getpid();printf(“\n the parent process ID is %d\n”, pid1);}else{pid2=getpid();printf(“\n the child process ID is %d\n”, pid2);}}SAMPLE OUTPUT:Cc fork.c./a.outTHE CHILD PROCESS ID IS 8640THE PARENT PROCESS ID IS 8644RESULT: The shell programs written using testing and loops were executed successfully.Ex.No:4C programs to simulate UNIX commands like cp, ls, grep.AIM: To write simple C programs to simulate UNIX commands like cp, ls, grep.1.Program for simulation of cp unix commands#include<fcntl.h>#include<unistd.h>#include<stdio.h>main(int argc,char *argv[]){FILE *fp;char ch;int sc=0;fp=fopen(argv[1],"r");if(fp==NULL) printf("unable to open a file",argv[1]);else{ while(!feof(fp)) { ch=fgetc(fp); if(ch==' ') sc++; } printf("no of spaces %d",sc); printf("\n"); fclose(fp); }}2.PROGRAM FOR SIMULATION OF LS UNIX COMMANDS#include<stdio.h>#include<dirent.h>main(int argc, char **argv){DIR *dp;struct dirent *link;dp=opendir(argv[1]);printf(“\n contents of the directory %s are \n”, argv[1]);while((link=readdir(dp))!=0)printf(“%s”,link->d_name);closedir(dp);}SAMPLE OUTPUT: Cc list.c./a.out osCONTENTS OF THE DIRECTORY OS AREPriority.cRobin.ccopy3. PROGRAM FOR SIMULATION OF GREP UNIX COMMANDS#include<stdio.h>#include<string.h>#define max 1024void usage(){printf(“usage:\t. /a.out filename word \n “);}int main(int argc, char *argv[]){FILE *fp;char fline[max];char *newline;int count=0;int occurrences=0;if(argc!=3){usage();exit(1);}if(!(fp=fopen(argv[1],”r”))){printf(“grep: couldnot open file : %s \n”,argv[1]);exit(1);}while(fgets(fline,max,fp)!=NULL){count++;if(newline=strchr(fline, ‘\n’))*newline=’\0’;if(strstr(fline,argv[2])!=NULL){printf(“%s: %d %s \n”, argv[1],count, fline);occurrences++;}}}SAMPLE OUTPUTCAT>SAMPONEONE TWOTHREE FOURCc grep.c./a.out samp oneSamp:1 oneSamp:2 one twoRESULT: The C programs to simulate UNIX commands like cp, ls, grep, etc. were executed successfully.