**Election Algorithm**:

**#include<stdio.h>**

**#include<stdlib.h>**

**struct process**

**{**

**int no;**

**int priority;**

**int active;**

**struct process \*next;**

**};**

**typedef struct process p;**

**struct priority**

**{**

**int pri;**

**struct priority \*next;**

**p \*pp;**

**};**

**typedef struct priority pri;**

**pri\* find\_priority(p \*head,pri \*head1)**

**{**

**p \*p1;**

**pri \*p2,\*p3;**

**p1=head;**

**while(p1->next!=head)**

**{**

**if(p1->active==1)**

**{**

**if(head1==NULL)**

**{**

**head1=(pri\*)malloc(sizeof(pri));**

**head1->pri=p1->priority;**

**head1->next=NULL;**

**head1->pp=p1;**

**p2=head1;**

**}**

**else**

**{**

**p3=(pri\*)malloc(sizeof(pri));**

**p3->pri=p1->priority;**

**p3->pp=p1;**

**p3->next=NULL;**

**p2->next=p3;**

**p2=p2->next;**

**}**

**p1=p1->next;**

**}**

**else**

**p1=p1->next;**

**}//end while**

**p3=(pri\*)malloc(sizeof(pri));**

**p3->pri=p1->priority;**

**p3->pp=p1;**

**p3->next=NULL;**

**p2->next=p3;**

**p2=p2->next;**

**p3=head1;**

**return head1;**

**}//endfind\_priority()**

**int find\_max\_priority(pri \*head)**

**{**

**pri \*p1;**

**int max=-1;**

**int i=0;**

**p1=head;**

**while(p1!=NULL)**

**{**

**if(max<p1->pri && p1->pp->active==1)**

**{**

**max=p1->pri;**

**i=p1->pp->no;**

**}**

**p1=p1->next;**

**}**

**return i;**

**}**

**void bully()**

**{**

**p \*head;**

**p \*p1;**

**p \*p2;**

**int n,i,pr,maxpri,a,pid,max,o;**

**char ch;**

**head=p1=p2=NULL;**

**printf("\nEnter how many process: ");**

**scanf("%d",&n);**

**for(i=0;i<n;i++)**

**{**

**printf("\nEnter priority of process %d: ",i+1);**

**scanf("%d",&pr);**

**printf("\nIs process with id %d is active ?(0/1) :",i+1);**

**scanf("%d",&a);**

**if(head==NULL)**

**{**

**head=(p\*)malloc(sizeof(p));**

**if(head==NULL)**

**{**

**printf("\nMemory cannot be allocated");**

**exit(0);**

**}**

**head->no=i+1;**

**head->priority=pr;**

**head->active=a;**

**head->next=head;**

**p1=head;**

**}**

**else**

**{**

**p2=(p\*)malloc(sizeof(p));**

**if(p2==NULL)**

**{ printf("\nMemory cannot be allocated");**

**exit(0);**

**}**

**p2->no=i+1;**

**p2->priority=pr;**

**p2->active=a;**

**p1->next=p2;**

**p2->next=head;**

**p1=p2;**

**}**

**}//end for**

**printf("\nEnter the process id that invokes election algorithm: ");**

**scanf("%d",&pid);**

**p2=head;**

**while(p2->next!=head)**

**{**

**if(p2->no==pid)**

**{**

**p2=p2->next;**

**break;**

**}**

**p2=p2->next;**

**}**

**printf("\nProcess with id %d has invoked election algorithm",pid);**

**printf("\nElection message is sent to processes");**

**while(p2->next!=head)**

**{**

**if(p2->no>pid) printf("%d",p2->no);**

**p2=p2->next;**

**}**

**printf("%d",p2->no);**

**p2=head;**

**max=0;**

**while(1)**

**{**

**if(p2->priority>max && p2->active == 1) max=p2->no;**

**p2=p2->next;**

**if(p2==head) break;**

**}**

**printf("\nProcess with the id %d is the co-ordinator",max);**

**while(1)**

**{**

**printf("\nDo you want to continue?(y/n): ");**

**scanf("%c",&ch);**

**if(ch=='n' || ch=='N') break;**

**p2=head;**

**while(1)**

**{**

**printf("\nEnter the process with id %d is active or not (0/1): ",p2->no);**

**scanf("%d",&p2->active);**

**p2=p2->next;**

**if(p2==head)**

**break;**

**}**

**printf("\nEnter the process id that invokes election algorithm: ");**

**scanf("%d",&pid);**

**printf("\nElection message is sent to processes ");**

**while(p2->next!=head)**

**{**

**if(p2->no>pid) printf("%d",p2->no);**

**p2=p2->next; } printf("%d",p2->no);**

**p2=head;**

**max=0;**

**while(1)**

**{**

**if(p2->no>max && p2->active==1) max=p2->no;**

**p2=p2->next;**

**if(p2==head) break;**

**}**

**printf("\nProcess with id %d is the co-ordinator",max);**

**}**

**} void main()**

**{**

**bully();**

**}**