**安徽科技学院**

**软件类实验报告**

|  |  |
| --- | --- |
| 课程名称： | 操作系统 |
| 学 号： | 2702200209 |
| 姓 名： | 乔梁 |
| 班 级： | 计算机科学与技术204 |
| 指导教师： |  |
| 开课学期： | 2020-2021-2 |
| 学 院： | 信息与网络工程学院 |

安徽科技学院教务处

|  |  |  |  |
| --- | --- | --- | --- |
| **实验名称** | **进程管理** | **实验学时** | **2** |
| **实验场所** | **专业实验室315** | **实验性质** | **验证** |
| **一、实验目标**：  1.理解进程的概念，明确进程和程序  的区别。  2.理解并发执行的实质  3.掌握进程的创建、睡眠、撤销等进程  的控制方法。 | | | |
| **二、实验原理：**  用C语言编写程序，模拟实现  ·创建新的进程;  ·查看运行进程;  ·换出某个进程;  ·杀死运行进程等功能。 | | | |
| **三、实验要求：**  用C语言编写程序，模拟实现  ·创建新的进程;  ·查看运行进程;  ·换出某个进程;  ·杀死运行进程等功能。 | | | |
| **四、实验步骤： （包括详细实验操作步骤、流程图、代码及重要语句注释）**  **流程图**    实验时对照代码提示操作即可。  #include<bits/stdc++.h>  using namespace std;  struct PCB {  int PID;  int privilege;  string size;  int status;  string informatin;  };  const int MAX\_PROCESS = 20;  struct PCB processInmemory[MAX\_PROCESS];//PCB队列中最多只能存在20个进程。  int totalProcess = 0, suspendProcess = 0, PID;  void create() {  int i, j;  if (totalProcess >= MAX\_PROCESS) {  cout << "内存己满,请先换出或杀死进程" << endl;  }  else {  for (i = 0; i < MAX\_PROCESS; i++) {  if (processInmemory[i].status == 0) break;//找到插入进程位置  }  cout << "请输入新进程pid" << endl;  cin >> processInmemory[i].PID;  for (j = 0; j < i; j++) {  if (processInmemory[i].PID == processInmemory[j].PID) {  cout << "该进程已存在" << endl;  return;  }  }  cout << "请输入新进程优先级" << endl;  cin >> processInmemory[i].privilege;  cout << "请输入新进程大小" << endl;  cin >> processInmemory[i].size;  cout << "请输入新进程内容" << endl;  cin >> processInmemory[i].informatin;  processInmemory[i].status = 1;  totalProcess++;  }  }  void run() {  int i;  int flag = 0;//内存中是否有进程  for (i = 0; i < totalProcess; i++) {  if (processInmemory[i].status == 1) {  cout << "pid= " << processInmemory[i].PID << "\t"  << "privilege= " << processInmemory[i].privilege << "\t"  << "size= " << processInmemory[i].size << "\t"  << "status= " << processInmemory[i].status << "\t"  << "informatin= " << processInmemory[i].informatin << endl;  flag = 1;  }  }  if (flag == 0) {  cout << "当前没有运行进程" << endl;  }  }  void kill() {  int flag = 0;//标记是否有对应PID的进程  if (totalProcess == 0) {  cout << "当前没有运行进程" << endl;  return;  }  cout << "输入杀死进程的 ID 值" << endl;  cin >> PID;  for (int i = 0; i < MAX\_PROCESS; i++) {  if (PID == processInmemory[i].PID) {  if (processInmemory[i].status == 1) {  processInmemory[i].status = 0;  totalProcess--;  cout << "已成功杀死进程" << endl;    }  else if (processInmemory[i].status == 0) {  cout << "要杀死的进程不存在" << endl;  }  else if (processInmemory[i].status == 2) {  cout << "要杀死的进程已被挂起" << endl;  }  flag = 1;  break;  }  }  if (flag == 0) {  cout << "要杀死的进程不存在" << endl;  }  }  void suspend() {  int flag = 0;//标记是否有需要挂起的进程  if (totalProcess == 0) {  cout << "当前没有运行进程" << endl;  return;  }  cout << "输入换出进程的ID值" << endl;  cin >> PID;  for (int i = 0; i < totalProcess; i++) {  if (PID == processInmemory[i].PID) {  if (processInmemory[i].status == 1) {  processInmemory[i].status = 2;  suspendProcess++;  cout << "已经成功换出进程" << endl;    }  else if (processInmemory[i].status == 0) {  cout << "要换出的进程不存在(程序还没启动)" << endl;  }  else if (processInmemory[i].status == 2) {  cout << "要换出的进程已被挂起" << endl;  }  flag = 1;  break;  }  }  if (flag == 0) {  cout << "要换出的进程不存在（pid不存在）" << endl;  }  }  void active() {  int flag = 0;//标记是否有需要激活的进程  if (totalProcess == 0) {  cout << "当前没有运行进程" << endl;  return;  }  if (suspendProcess == 0) {  cout << "当前没有挂起进程" << endl;  return;  }  cout << "输入pid:" << endl;  cin >> PID;  for (int i = 0; i < MAX\_PROCESS; i++) {  if (PID == processInmemory[i].PID) {  if (processInmemory[i].status == 2) {  processInmemory[i].status = 1;  suspendProcess--;  cout << "已经成功激活进程" << endl;  }  else if (processInmemory[i].status == 0) {  cout << "要激活的进程不存在" << endl;  }  else if (processInmemory[i].status == 1) {  cout << "要激活的进程已被激活" << endl;  }  flag = 1;  break;  }  }  if (flag == 0) {  cout << "要激活的进程不存在" << endl;  }  }  void initiateProcess() {  for (int i = 0; i < MAX\_PROCESS; i++)  {  processInmemory[i].status = 0;  }  }  void menu() {  cout << "\*进程演示系统\*" << endl;  cout << "1.创建新的进程 2.查看运行进程" << endl;  cout << "3.换出某个进程 4.杀死运行进程" << endl;  cout << "5.激活某个进程 6.退出系统" << endl;  cout << "请选择(1~6)" << endl;  }  int main() {  int choice;  while (true) {  system("cls");  menu();  cin >> choice;  switch (choice) {  case 1:  create();  break;  case 2:  run();  break;  case 3:  suspend();  break;  case 4:  kill();  break;  case 5:  active();  break;  case 6:  cout << "成功退出系统" << endl;  system("pause");  return 0;  }  system("pause");  }  return 0;  } | | | |
| **五、实验结果：** | | | |
| **六、实验小结：**  通过本次实验，我利用C语言程序熟悉了计算机中的程序管理操作，明白了进程在计算机中的创建，睡眠和撤销，对并发运行的实质有了更深的了解。 | | | |
|  | | | |