**fbMÔN HỌC: HỆ ĐIỀU HÀNH**

**CÂU HỎI VÀ BÀI TẬP CHƯƠNG 3**

BIÊN SOẠN: PHAN ĐÌNH DUY

1. Một tiến trình chứa những thành phần gì?

giải

Các thành phần của 1 tiến trình:

- Text section

- Data sectuib

- Program counter, processor registers

- Heap section

- Stack section

1. Tiến trình có những trạng thái nào? Cách tiến trình chuyển trạng thái?

giải

Những trạng thái của tiến trình: new, ready, running, waiting, terminated

Cách chuyển trạng thái: nhìn chung là nhờ vào 1 sự kiện. Chẳng hạn:

ready được cấp CPU sẽ chuyển sang running

running chạy đến phần cần nhập xuất sẽ chuyển sang waiting

Nhập xuất xong thì waiting chuyển sang ready

chạy xong thì running chuyển sang terminated

1. Tại sao phải cộng tác giữa các tiến trình?
2. PCB là gì? Dùng để làm gì?
3. Tiểu trình là gì?
4. Trình tự thực thi của tiến trình cha và tiến trình con?
5. (Bài tập mẫu) Cho đoạn chương trình sau:

|  |
| --- |
| int main (int argc, char\*\* argv)  {  int i = 2;  while (i < =5)  {  i++;  if (i % 2 == 0)  {  printf (“Hello”);  printf (“Hi”);  }  else  {  printf (“Bye”);  }  }  exit (0);  } |

Hỏi trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái nào? Vẽ sơ đồ chuyển trạng thái trong quá trình thực thi?

Trả lời:

Trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái như sau: new – ready – running – waiting – ready – running – waiting – ready – running – waiting – ready – running – waiting – ready – running – terminated

1. Cho đoạn chương trình sau:

|  |
| --- |
| /\* test.c \*/  int main(int argc, char\*\* argv)  {  int a;  for (int i = 1; i < 5; i++)  {  if (i % 2 == 0)  printf(“Hello world\n");  else a = 5\*9;  }  exit(0);  } |

Hỏi trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái nào? Vẽ sơ đồ chuyển trạng thái trong quá trình thực thi?

giải

Trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái như sau: new – ready – running – waiting – ready – running – waiting - ready – running – waiting – ready – running - terminated

1. Cho đoạn chương trình sau:

|  |
| --- |
| int main (int argc, char\*\* argv)  {  int i = 2;  while (i < =5)  {  i++;  if (i % 2 == 0)  {  printf (“Hello”);  printf (“Hi”);  }  else  {  printf (“Bye”);  }  }  exit (0);  } |

Hỏi trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái nào? Vẽ sơ đồ chuyển trạng thái trong quá trình thực thi?

**giải**

Trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái như sau: new – ready – running – waiting – ready – running – waiting – ready – running – waiting – ready – running – waiting – ready – running – terminated

1. Cho đoạn chương trình sau:

|  |
| --- |
| int main (int argc, char\*\* argv)  {  int a, b, I;  for (I = 16, i >=6; i --)  {  if (i % 3 == 0)  {  printf (“Số %d chia hết cho 3”, i);  }  else  {  a = b + i;  }  }  exit (0);  } |

Hỏi trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái nào? Vẽ sơ đồ chuyển trạng thái trong quá trình thực thi?

**giải**

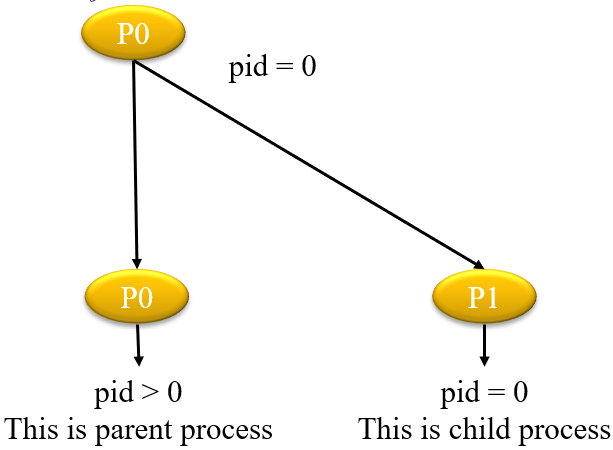
Trong đề thì biến i chưa khai báo, nhưng có thể hiểu I và i là 1. Trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái như sau: new – ready – running – waiting – ready – running –waiting – ready – running – waiting – ready – running – waiting – ready – running – terminated.

1. (Bài tập mẫu) Cho đoạn code sau, hỏi khi chạy, bao nhiêu process được sinh ra và chương trình sẽ in ra những gì? Vẽ cây tiến trình khi thực thi đoạn chương trình sau

|  |
| --- |
| #include <stdio.h>  #include <unistd.h>  int main (int argc, char \*argv[]){  int pid;  /\* create a new process \*/  pid = fork();  if (pid > 0){  printf(“This is parent process”);  wait(NULL);  exit(0);}  else if (pid == 0) {  printf(“This is child process”);  execlp(“/bin/ls”, “ls”, NULL);  exit(0);}  else { // pid < 0  printf(“Fork error\n”);  exit(-1);  }  } |

Trả lời:

Khi chạy đoạn chương trình trên, khi chạy hết sẽ có 2 process được sinh ra bao gồm 1 tiến trình cha và 1 tiền trình con. Theo chương trình trên thì tiến trình cha sẽ in ra dòng chữ “This is parent process”; và tiến trình con sẽ in ra dòng chữ “This is child process”. Cây tiến trình khi thực thi đoạn chương trình trên như sau:



1. Cho đoạn code sau, hỏi khi chạy, bao nhiêu process (kể cả cha) được sinh ra? Vẽ cây tiến trình khi thực thi đoạn chương trình sau

|  |
| --- |
| int main()  {  fork();  fork();  fork();  fork();  return 0;  } |

**giải**

Số process sinh ra: 16. P0 là tiến trình cha, các tiến trình còn lại là con.

Để thuận tiện, đặt tên cho 4 lần gọi fork() là f0, f1, f2, f3.

Chart, funnel chart

Description automatically generated

1. Cho đoạn code sau, hỏi khi chạy thì tiến trình được tạo ra từ chương trình trên sẽ in ra màn hình những gì? Vẽ cây tiến trình và những từ được in ra khi thực thi đoạn chương trình sau?

|  |
| --- |
| #include <stdio.h>  #include <unistd.h>  int main()  {  int i;  for (i = 0; i < 4; i++)  {  fork();  printf("hello\n");  }  return 0;  } |

**giải**

Khi chạy đoạn chương trình trên, khi chạy hết sẽ có 16 process được sinh ra bao gồm 1 tiến trình cha và 15 tiền trình con. Theo chương trình trên thì tiến trình cha và tiến trình con sẽ in ra dòng chữ “hello”. Cây tiến trình khi thực thi đoạn chương trình trên như sau:

Timeline

Description automatically generated

1. Cho đoạn code sau, hỏi khi chạy thì tiến trình được tạo ra từ chương trình trên sẽ in ra màn hình những gì? Vẽ cây tiến trình và những từ được in ra khi thực thi đoạn chương trình sau?

|  |
| --- |
| int main (int argc, char \*\*argv)  {  int pid;  printf(“Tiến trình cha \n”);  pid = fork();  if (pid > 0)  {  fork();  printf(“Tiến trình cha \n”);  }  else  {  printf(“Tiến trình con \n”);  if(fork() > 0 )  printf(“Tiến trình cha \n”);  else  printf(“Tiến trình con \n”);  }  } |

**giải**

Khi chạy đoạn chương trình trên, màn hình sẽ in ra 4 lần “tiến trình cha” và 2 lần “tiến trình con”.

Gọi các lần xuất hiện của else trong đoạn code là else 1, else 2. Vẽ cây tiến trình và những từ được in ra khi thực thi đoạn chương trình:

Graphical user interface, text, application

Description automatically generated

1. Cho đoạn code chương trình sau:

|  |
| --- |
| if (fork() == 0)  {     a = a + 5;     printf("%d,%d\n", a, &a);  }  else  {      a = a –5;      printf("%d, %d\n", a, &a);  } |

Giả sử u, v là các giá trị được in ra bởi process cha, và x, y là các giá trị được in ra bởi process con. Tìm mối quan hệ giữa u, v và x, y?

giải

giả sử u, v lần lượt là a, &a của process cha, x, y lần lượt là a, &a của process con.

Nhận xét:

v và y bằng nhau.

x = u + 10.

giải thích:

v và y bằng nhau vì tiến trình con chỉ sử dụng lại biến a của tiến trình cha. Nhưng khi tiến trình con kết thúc, memory sẽ được giải phóng, trong đó có cả giá trị của biến a, khiến a trở lại thành giá trị như trước khi vào tiến trình con. Khi tới tiến trình cha, a = a – 5. Giá trị xuất ra gọi là u. Vậy x = 10 + u.