我建立了一個struct來建置share memory,  
typedef struct region{

int child\_pid,

guess,

parent1,

parent2,

child1,

child2,

num\_parent,

num\_child,

bomb\_parent,

bomb\_child,

winner,

statue;

}REGION; -> 最下面會詳細說明

建完以後fork()出子程式,

parent的seed會等於argv[1],

child的seed會等於argv[2]，

seed是int， argv是char\* 所以我是用atoi (char to int)

ex:seed = atoi (argv[1])，

parent先輸出randomseed，然後將statue設為2， 使parent暫停，

換child輸出randomseed，然後將statue設為1，使child暫停，

之後parent跟child創造gunboat的第一個位置，我是用rand() % 16，也就是0~15 -> 最下面會詳細說明

然後第二個位置，我是用第一個位置去判斷 -> 最下面會詳細說明

創造完後，

parent先輸出第一個和第二個位置，然後將statue設為2， 使parent暫停，

換child輸出第一個和第二個位置，然後將statue設為1，使child暫停，

之後就換兩個互猜，輸出轟炸結果，最後parent輸出誰贏並用了幾顆炸彈，那我是分別在parent和child各有一個迴圈

->parent的迴圈(最先執行)

while (num\_child != 0 && num\_parent != 0){

parent的炸彈次數+1

guess會等於rand()%16 (0~15)

然後我有一維陣列，如果那個位置沒有被轟過，值為0，有的話，值為1，藉此避免重複轟炸同一個位置。

parent輸出 (guess/4, guess%4)

之後statue設成2，使parent暫停 ->換child執行

會先判斷num\_child是不是等於0，

等於0: break

不等於0:

再判斷guess跟parent的兩個位置有無相等

----------------------------------------------------------

有: num\_parent-1，

值等於0，parent輸出hit and sinking，winner設為1，然後break

值等於1，parent輸出hit ，被轟炸的位置(parent1或parent2)射成16，避免再次hit到

沒有: parent輸出missed

}

->child的迴圈(parent迴圈暫停，換child執行)

while (num\_child != 0 && num\_parent != 0){

先判斷guess跟child的兩個位置有無相等

----------------------------------------------------------

有: num\_child-1，

值等於0，child輸出hit and sinking，winner設為2，然後將 statue設成1，使child暫停，然後break ->換parent執行

值等於1，child輸出hit ，被轟炸的位置(child1或child2)射成16，避免再次hit到

沒有: child輸出missed

----------------------------------------------------------

child的炸彈次數+1

guess會等於rand()%16 (0~15)

然後我有一維陣列，如果那個位置沒有被轟過，值為0，有的話，值為1，藉此避免重複轟炸同一個位置。

child輸出 (guess/4, guess%4)

然後將 statue設成1，使child暫停 ->換parent執行

}

兩個迴圈都結束在parent或child其中一個兩個位置都被轟炸過，之後parent輸出贏的那個pid那我是先用winner的值判斷誰輸誰贏，之後在parent輸出winner用的炸彈數量。

最後釋放掉share memory

有做到從命令列讀入整數參數並設定乘亂數種子,

parent用fork產生出一個子行程,

有用到share memory互相傳遞且子行程有正確計算比較結果,但是用空迴圈,getpid()能正常運作

執行方式:

在終端機先編譯檔案

gcc s1091447 -o s1091447

執行

./hw (0 <= seed <= 99的亂數種子) (0 <= seed <= 99的亂數種子) (0 or 1)

ex: ./s1091447 10 5 1

變數名稱說明

|  |  |
| --- | --- |
| child\_pid | 儲存child的pid |
| guess | 儲存猜測的位置  (0~15) -> 之後說明 |
| parent1 | parent' s gunboat的第一個位置 |
| parent2 | parent' s gunboat的第二個位置 |
| child1 | child' s gunboat的第一個位置 |
| child2 | child' s gunboat的第二個位置 |
| num\_parent | 計算parent的gunboat還剩餘的數目(預設2) |
| num\_child | 計算child的gunboat還剩餘的數目(預設2) |
| bomb\_parent | 計算parent炸的次數(預設0) |
| bomb\_child | 計算child炸的次數(預設0) |
| winner | 1:child,2:parent |
| statue | 1:child暫停,2:parent暫停 |

位置說明 4\*4

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |
| (0/4,0%4)=  (0,0) | (1/4,1%4)=  (0,1) | (2/4,2%4)=  (0,2) | (3/4,3%4)=  (0,3) |
| 4 | 5 | 6 | 7 |
| (4/4,4%4)=  (1,0) | (5/4,5%4)=  (1,1) | (6/4,6%4)=  (1,2) | (7/4,7%4)=  (1,3) |
| 8 | 9 | 10 | 11 |
| (8/4,8%4)=  (2,0) | (9/4,9%4)=  (2,1) | (10/4,10%4)=  (2,2) | (11/4,11%4)=  (2,3) |
| 12 | 13 | 14 | 15 |
| (12/4,12%4)=  (3,0) | (13/4,13%4)=  (3,1) | (14/4,14%4)=  (3,2) | (15/4,15%4)=  (3,3) |

第二個位置 為第一個位置+4或+1或-1或-4

|  |  |  |
| --- | --- | --- |
|  | -4 |  |
| -1 | 第一個位置 | +1 |
|  | +4 |  |

所以我是用迴圈去寫

while(true)

如果rand%16()的值不是第一個位置+4或+1或-1或-4就一直重複，

是的話，就break