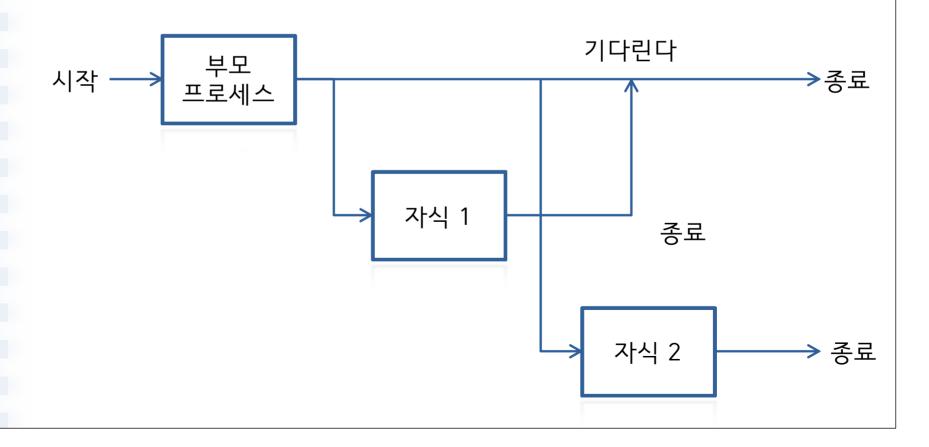
OS 프로그래밍 과제 #1 waitpid()로 기다리기 홀수 학번 문제

운영체제, 임철수 교수님 2012305087 최찬휘 2015-1학기

문제

자식 프로세스 두 개를 만들되
 첫 번째 프로세스의 종료를 기다리고 종료



주어진 스켈레톤 코드 (1/2)

- #include
 - stdio.h: C언어 표준 입출력
 - -sys/types.h: 기본 시스템 자료형
 - -sys/wait.h : 대기 (wait)
 - 자식 프로세스의 상태변화를 대기
 - wait(), waitpid(), waitid()
 - -unistd.h: 표준 POSIX API

주어진 스켈레톤 코드 (2/2)

- 없는 부분
 - 자식 프로세스의 pid 출력
 - 부모 프로세스의 작업완료 출력

```
run child-2[pid: 8533]
run child-1[pid: 8532]
child [pid: 8532] terminated with code
```

- -부모 프로세스의 대기 동작
 - → waitpid() 함수 사용
- -자식 프로세스의 값 200 반환 부분
 - 미처 고려하지 못함

waitpid() [1/3]

- 자식 프로세스가 종료될 때까지 대기
- waitpid(int pid, int* status, int options)
 - int pid
 - < -1 : (프로세스) 그룹 ID가 |pid| 인 것들의 종료 대기
 - -1 : 모든 자식 프로세스의 종료 대기
 - 0 : (프로세스) 그룹 ID가 같은 모든 자식 프로세스의 종료 대기
 - > 0 : 프로세스 ID가 같은 자식 프로세스의 종료 대기

waitpid() [2/3]

- waitpid(int pid, int* status, int options)
 - -int* status
 - 자식 프로세스의 종료 상태 정보
 - 각종 매크로로 확인 가능
 - 주소가 아니라 status에 쓰인 변수 자체를 사용
 - WIFEXITED(status) : 정상 종료시 true 반환
 - WEXITSTATUS(status): 정상 종료시 종료 상태를 반환
 - WIFSIGNALED(status) : 자식 프로세스가 신호를 받아...
 - ...생략. 터미널에 man waitpid 를 입력하세요

waitpid() [3/3]

- waitpid(int pid, int* status, int options)
 - int options
 - 다양한 옵션 상수를 OR 하여 적용 가능
 - 0: 자식 프로세스 종료까지 대기
 - WNOHANG: 자식 프로세스의 실행 여부만 확인, 바로 복귀: block 되지 않음

fork() / getpid()

• fork() 의 리턴값

-0: 자식인 경우

-> 0: 부모인 경우: 생성된 자식 프로세스의 pid

--1: 생성 실패

RETURN VALUE

On success, the PID of the child process is returned in the parent, and 0 is returned in the child. On failure, -1 is returned in the parent, no child process is created, and errno is set appropriately.

getpid(): 현재 프로세스의 pid 반환

DESCRIPTION

getpid() returns the process ID of the calling process. (This is often used by routines that generate unique temporary filenames.)

getppid() returns the process ID of the parent of the calling process.

핵심 부분

• 부모 프로세스에서 자식 1, 자식 2를 생성 후 waitpid() 로 대기

```
/* pid 1 종료 대기 */
int status1;
waitpid(pid1, &status1, 0);
printf("child [pid: %d] terminated with code %d\n",
pid1, status1);
```

• status1이 아닌 WEXITSTATUS(status1) 을 했었어야

- 변수 선언 위치가 편의상 중간이므로 ANSI C 와는 비호환입니다. 주의바랍니다. (C99 이상)
- 결과를 검증하기 위해 몇몇 출력이 더 포함되어 있습니다.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
                                                                         perror<del>는</del>
#include <unistd.h>
                                                                         프로세스
int main(){
                                                                         에러문 출력
  printf("parent-process[pid: %d] : start\n", getpid());
  int pid1;
  /* 첫 번째 자식 프로세스 생성 */
                                               lubuntu@lubuntu= /irtualBox:~/Documents/osLec2015/report/1$ ./fork1
  pid1 = fork();
                                               parent-process[pid: 8531] : start
                                              parent-process[pid: 8531] : launched child-1 parent-process[pid: 8531] : launched child-2
  if (pid1 == -1)
    perror("fork failed, child1\n");
                                               run child-2[pid: 8533]
                                               run child-1[pid: 8532]
                                               child [pid: 8532] terminated with code 0
  /* 부모 프로세스 Phase 1 */
                                               lubuntu@lubuntu-VirtualBox:~/Documents/osLec2015/report/1$
  else if (pid1 != 0) {
```

```
/* 부모 프로세스 Phase 1 */
      else if (pid1 != 0) {
17
18
        printf ("parent-process[pid: %d] : launched child-1\n",
19
        getpid());
20
        /* 두 번째 자식 프로세스 생성 */
21
        int pid2;
22
        pid2 = fork();
23
        if (pid2 == -1)
24
25
          perror("fork failed, child2\n");
26
27
        /* 부모 프로세스 Phase 2 */
        else if (pid2 != 0) {
```

```
lubuntu@lubuntu-Virtua Box:~/Documents/osLec2015/report/1$ ./fork1
parent-process[pid: 8531] : launched child-1
parent-process[pid: 8531] : launched child-1
run child-2[pid: 8533]
run child-1[pid: 8532]
child [pid: 8532] terminated with code 0
lubuntu@lubuntu-VirtualBox:~/Documents/osLec2015/report/1$
```

```
/* 부모 프로세스 Phase 2 */
         else if (pid2 != 0) {
28
29
           printf ("parent-process[pid: %d] : launched
30
           child-2\n", getpid());
           /* pid 1 종료 대기 */
32
33
           int status1;
           waitpid(pid1, &status1, 0);
34
           printf("child [pid: %d] terminated with code %d\n",
35
           pid1, status1);
36
         /* 자식 프로세스 2 */
37
         else {
                                              lubuntu@lubuntu-VirtualBox:~/Documents/osLec2015/rep_rt/1$ ./fork1
                                              child [pid: 8532] terminated with code 0
```

```
else 에서
                                                                         return 200;
         /* 자식 프로세스 2 */
                                                                         을 했었어야
38
         else +
           printf("run child-2[pid: %d]\n", getpid());
39
40
41
42
       /* 자식 프로세스 1 */
43
44
      else {
        printf("run child-1[pid: %d]\n", getpid());
         sleep(1);
47
       return 0:
                                            ubuntu@lubuntu-VirtualBox:~/Documents/osLec2015/report/1$ ./fork1
                                            parent-process[pid: 8531] : start
        자식 프로세스의
                                            parent-process[pid: 8531] : launcled child-
                                                               : launched child-
        종료를 기다리는지
        확인하기 위해
                                            cniia [pia: 8532] terminated with code 0
                                            lubuntu@lubuntu-VirtualBox:~/Documents/osLec2015/report/1$
        일부러 1초 대기
        ( sleep(1); )
```

실행 결과

- 정상적으로 child-1 종료를 대기함.
- child-1 보다 child-2 가 먼저 출력
 - -pid 값은 child-2가 더 큼
 - 생성 순서는 흐름상 child-1이 우선이지만 child-2가 먼저 출력
 - **-**→ ?????

```
lubuntu@lubuntu-VirtualBox:~/Documents/osLec2015/report/1$ ./fork1
parent-process[pid: 8531] : start
parent-process[pid: 8531] : launched child-1
parent-process[pid: 8531] : launched child-2
run child-2[pid: 8533]
run child-1[pid: 8532]
child [pid: 8532] terminated with code 0
lubuntu@lubuntu-VirtualBox:~/Documents/osLec2015/report/1$
```