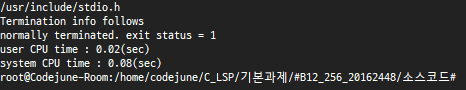
컴퓨터학부 20162448 김병준

1. 결과



1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <errno.h>  #include <sys/resource.h>  #include <sys/wait.h>  double ssu\_maketime(struct timeval \*time);  void term\_stat(int stat);  void ssu\_print\_child\_info(int stat, struct rusage \*rusage);  int main(void){  struct rusage rusage;  pid\_t pid;  int status;  if((pid = fork()) == 0){ // 자식 프로세스 생성, stdio.h를 찾는 함수  char \*args[] = {"find", "/", "-maxdepth", "4", "-name", "stdio.h", NULL}; // 인자 배열  if(execv("/usr/bin/find", args) < 0){ // 자식프로세스는 find명령어와 인자들을 넘겨서 실행하고 자신은 종료  fprintf(stderr, "exec error\n");  exit(1);  }  }  if(wait3(&status, 0, &rusage) == pid) // 자식프로세스가 종료되면  ssu\_print\_child\_info(status, &rusage); // 실행  else { // 비정상의 경우  fprintf(stderr, "wait3 error\n");  exit(1);  }  exit(0);  }  double ssu\_maketime(struct timeval \*time){ // 시스템 & 유저 시간 리턴  return ((double)time -> tv\_sec + (double)time -> tv\_usec/1000000.0);  }  void term\_stat(int stat){  if(WIFEXITED(stat)) // 정상적 종료  printf("normally terminated. exit status = %d\n", WEXITSTATUS(stat));  else if (WIFSIGNALED(stat)) // 시그널이 존재하는 종료  printf("abnormal termination by signal %d, %s\n", WTERMSIG(stat),  #ifdef WCOREDUMP  WCOREDUMP(stat)?"core dumped":"no core"  #else  NULL  #endif  );  else if (WIFSTOPPED(stat)) // 비정상종료  printf("stopped by signal %d\n", WSTOPSIG(stat));  }  void ssu\_print\_child\_info(int stat, struct rusage \*rusage){//USER ,SYSTEM CPU 출력  printf("Termination info follows\n");  term\_stat(stat); // 프로세스 종료 반환값 출력  printf("user CPU time : %.2f(sec)\n", ssu\_maketime(&rusage->ru\_utime));  printf("system CPU time : %.2f(sec)\n", ssu\_maketime(&rusage->ru\_stime));  } |