**MPI 예시 프로그램**

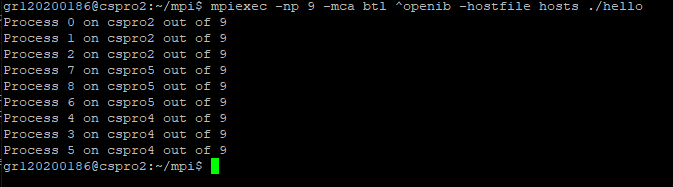
1. cspro2 접속
2. vim hello.c 로 아래의 프로그램을 작성

|  |
| --- |
| #include <stdio.h>  #include <mpi.h>  int main(int argc, char \*argv[]) {  int numprocs, rank, namelen;  char processor\_name[MPI\_MAX\_PROCESSOR\_NAME];  MPI\_Init(&argc, &argv);  MPI\_Comm\_size(MPI\_COMM\_WORLD, &numprocs);  MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);  MPI\_Get\_processor\_name(processor\_name, &namelen);  printf("Process %d on %s out of %d \n", rank, processor\_name, numprocs);  MPI\_Finalize();  } |

1. vim hosts 로 아래의 파일을 작성

|  |
| --- |
| cspro2.sogang.ac.kr slots=3  cspro4.sogang.ac.kr slots=3  cspro5.sogang.ac.kr slots=3 |

1. mpicc hello.c -o hello
2. mpiexec -np 9 -mca btl ^openib -hostfile hosts ./hello



**RPC 예시 프로그램**

1. cspro1에 접속
2. vim rand.x 로 아래의 파일을 작성

|  |
| --- |
| program RAND\_PROG {  version RAND\_VERS {  void INITIALIZE\_RANDOM(long) = 1  double GET\_NEXT\_RANDOM(void) = 2  } = 1  } = 0x31111111 |

1. rpcgen –C –a rand.x
2. vim rand\_client.c 로 기존의 내용을 삭제하고 아래의 프로그램을 작성

|  |
| --- |
| #include <stdlib.h>  #include <stdio.h>  #include "rand.h"  void main(int argc, char \*argv[ ]) {  int iters,i;  long myseed;  CLIENT \*clnt;  void \*result\_1;  double \*result\_2;  char \*arg;  if ( argc != 4 ) {  fprintf(stderr, "Usage: %s host seed iterations\n", argv[0]);  exit(1);  }  clnt = clnt\_create(argv[1], RAND\_PROG, RAND\_VERS, "udp");  if ( clnt == (CLIENT \*) NULL ) {  clnt\_pcreateerror(argv[1]);  exit(1);  }  myseed = (long) atoi (argv[2]);  iters = atoi (argv[3]);  result\_1 = initialize\_random\_1(&myseed, clnt);  if (result\_1 == (void \*) NULL) {  clnt\_perror(clnt, "call failed");  }  for( i=0; i<iters; i++) {  result\_2 = get\_next\_random\_1((void \*)&arg, clnt);  if (result\_2 == (double \*) NULL) {  clnt\_perror(clnt, "call failed");  }  else {  printf("%d : %f\n", i, \*result\_2);  }  }  clnt\_destroy(clnt);  exit(0);  } |

1. vim rand\_server.c 로 기존의 내용을 삭제하고 아래의 프로그램을 작성

|  |
| --- |
| #include <stdlib.h>  #include "rand.h"  void \*  initialize\_random\_1\_svc(long \*argp, struct svc\_req \*rqstp) {  static char \*result;  srand48(\*argp);  result = (void \*) NULL;  return (void \*) &result;  }  double \*  get\_next\_random\_1\_svc(void \*argp, struct svc\_req \*rqstp){  static double result;  result = drand48();  return (&result);  } |

1. makefile –f Makefile.rand
2. cspro2에서 ./rand\_server 를 실행

예시에서 서버 프로그램은 특별한 출력을 하지 않고 계속 클라이언트를 대기한다.

1. cspro1에서 ./rand\_client cspro2.sogang.ac.kr 20 10 를 실행

