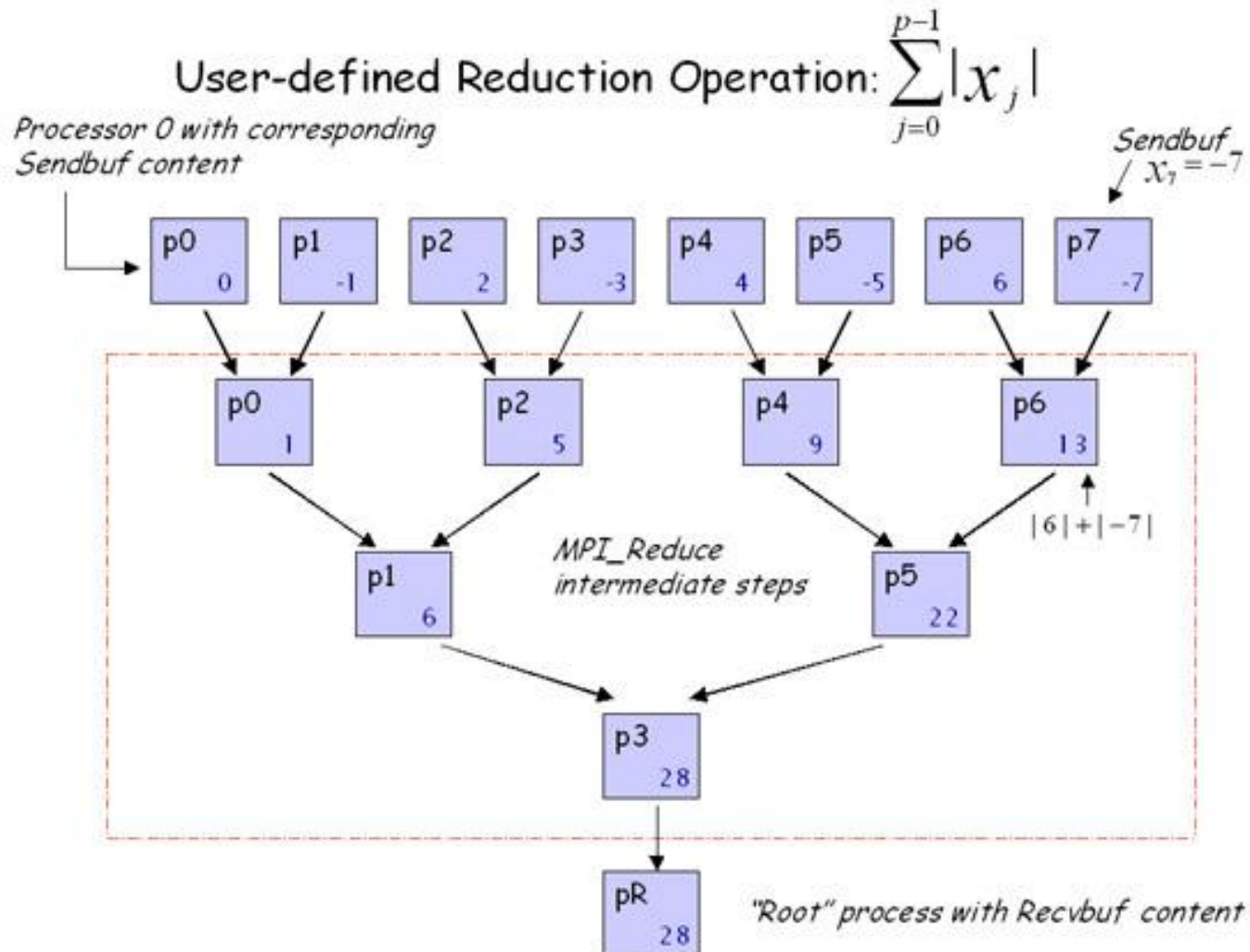


Parallel programming

MPI Interface

Create reduce operation



Create reduce operation

- Step 1: Implement your own reduction operator.

```
void onenorm(float *in, float *inout, int *len, MPI_Datatype
*type)
{
    int i;
    for (i=0; i<*len; i++)
    {
        *inout = fabs(*in) + fabs(*inout);
        in++;
        inout++;
    }
}
```

Create reduce operation

- Step 2: Register the function **onenorm** with MPI by:

MPI_Op_create(onenorm, commune, myop);

- **onenorm** is the function to be registered.
- **commune** is an input int (scalar) variable which takes on either **0** if the function satisfies the commutative property. Otherwise, set **commune** to **1**.
- **myop** is an int (output) to be used in subsequent reduction operation call in place of **onenorm**.

Create reduce operation

- Step 3: Call MPI reduction operation function **MPI_Reduce** using the registered **myop**.
- `MPI_Reduce(&sendbuf, &recvbuf, 1, MPI_FLOAT, myop, root, MPI_COMM_WORLD);`

12 - Example - Opp_mpi.c

15 - Kvazi šah

Treba napraviti Kvazi šahovsku igru. Igra se odvija na kvadratnoj tabli dimenzija $m \times m$. U igri učestvuju tri vrste figura: top, lovac i piun. Top se kreće kao i u standardnoj šahovskoj igri, tj. vertikalno ili horizontalno za proizvoljan broj mesta. Lovcu je dozvoljena kretnja samo po jedno mesto dijagonalno od mesta na kome se trenutno nalazi, a piunu samo po jedno mesto vertikalno ili horizontalno. Top je jači od lovca, lovac je jači od piuna, a piun je jači od topa. Kada se dve različite figure nađu na istom polju, ostaje ona koja je jača. Ukoliko neka figura odigra na polje gde se već nalazi figura istog tipa, ta figura jede dolazeću figuru, tj. uzima se da je jača jer brani svoju teritoriju. Figure se proizvoljno kreću po tabli u zavisnosti od svog tipa, i ne mogu odigrati potez van table. Ukoliko figura ipak izabere da se kreće van table, ostaje u mestu. Igra se završava kada na tabli ostane jedna figura, i ta figura je označena kao pobednik.

15 - Kvazi šah

U MPI programu prvo kreirati strukture:

- **Figura:**

- int id – id figure,
- int x – globalna x koordinata figure,
- int y – globalna y koordinata figure,
- int tip – tip figure, 0-top, 1-lovac, 2-piun,
- int brojPojedenih – koliko je figura pojela drugih figura,
- int brojPoteza – broj odigranih poteza figure, i

- **Polje:**

- int id – id figure koja se nalazi na polju (-1 označava da je polje prazno),
- int tip – tip figure koja se nalazi na datom polju.

15 - Kvazi šah

Učitati dimenziju table m . Smatrati da je m deljivo brojem procesa. Učitati i broj figura na svakom procesu. Inicijalno, svaki proces treba da postavi sve figure na delu table za koji je zadužen, vodeći računa o tome da se na jednom polju može naći samo jedna figura. Koordinate x i y figure su **globalne**. U svakom koraku igre svaka figura odigra po jedan potez sve dok se igra ne završi, odnosno dok ne ostane jedna figura na tabli. Odštampati koja je figura pobednik i koliko je ta figura pojela drugih figura.

15 - Kvazi šah

Saveti:

- Napraviti 3 funkcije (za svaki tip figure) koje određuju globalne koordinate na kojima će se figura nalaziti u sledećem trenutku u zavisnosti od trenutne pozicije i veličine table, jer figura ne sme napustiti tablu.
- Ukoliko se odlučite za pravljenje funkcije koja briše određenu figuru iz niza, možete samo na mestu date figure iskopirati poslednju figuru u nizu, i smanjiti veličinu niza za 1.
- Alocirajte dovoljno memorije, da ne morate realocirati tokom programa.
- Potrebna komunikacija se svodi samo na razmenu figura za koje su pojedinačni procesi zaduženi, a ne na razmenu polja.
- Kod kretanje topa, prvo je potrebno izabrati proizvoljno da li se kreće vertikalno ili horizontalno, a potom na kom polju treba da se nađe u sledećem koraku.

15 - Kvazi šah

- Top i lovac imaju četiri mogućnosti za kretnju, pa tako treba proizvoljno izabrati pravac u kome će se oni kretati.
- Lakše je prvo skinuti figure sa polja, odrediti nove pozicije svih figura, iskomunicirati sa ostalim procesima, i onda poređati ponovo figure na polja i izvršiti jedenje figura od strane dugih na istim poljima. Tako će na kraju koraka na svakom polju biti jedna ili nijedna figura.
- Voditi računa o komunikaciji da ne bi doslo do *deadlock*-a. Savet je da kada komunicira sa procesima levo od aktuelnog prvo prima pa šalje, a sa procesima desno od aktuelnog suprotno.