

APPENDIX A

IMPROVED CENTRALIZED MANAGER ALGORITHM

CentralManager

Read fault handler:

```

Lock( PTable[ p ].lock );
IF I am manager THEN
    receive page p from owner[ p ];
ELSE
    ask manager for read access to p and a copy of p;
PTable[ p ].access := read;
Unlock( PTable[ p ].lock );

```

Read server:

```

Lock( PTable[ p ].lock );
IF I am owner THEN BEGIN
    PTable[ p ].copyset
        := PTable[ p ].copyset  $\cup$  {RequestNode};
    PTable[ p ].access := read;
    send p;
END
ELSE IF I am manager THEN BEGIN
    Lock( ManagerLock );
    forward request to owner[ p ];
    Unlock( ManagerLock );
END;
Unlock( PTable[ p ].lock );

```

Write fault handler:

```

Lock( PTable[ p ].lock );
IF I am manager THEN
    receive page p from owner[ p ];
ELSE
    ask manager for write access to p and p's copyset;

    Invalidate( p, PTable[ p ].copyset );
    PTable[ p ].access := write;
    PTable[ p ].copyset := {};
    Unlock( PTable[ p ].lock );

```

Write server:

```

Lock( PTable[ p ].lock );
IF I am owner THEN BEGIN
    send p and PTable[ p ].copyset;
    PTable[ p ].access := nil;
END

```

```

ELSE IF I am manager THEN BEGIN
  Lock( ManagerLock );
  forward request to owner[ p ];
  owner[ p ] := RequestNode;
  Unlock( ManagerLock );
END;
Unlock( PTable[ p ].lock );

```

APPENDIX B

BROADCAST DISTRIBUTED MANAGER ALGORITHM

BroadcastManager

Read fault handler:

```

Lock( PTable[ p ].lock );
broadcast to get p for read;
PTable[ p ].access := read;
Unlock( PTable[ p ].lock );

```

Read server:

```

Lock( PTable[ p ].lock );
IF I am owner THEN BEGIN
  PTable[ p ].copyset :=
    PTable[ p ].copyset  $\cup$  [ RequestNode ];
  PTable[ p ].access := read;
  send p;
END;
Unlock( PTable[ p ].lock );

```

Write fault handler:

```

Lock( PTable[ p ].lock );
broadcast to get p for write;
Invalidate( p, PTable[ p ].copyset );
PTable[ p ].access := write;
PTable[ p ].copyset := {};
PTable[ p ].owner := self;
Unlock( PTable[ p ].lock );

```

Write server:

```

Lock( PTable[ p ].lock );
IF I am owner THEN BEGIN
  send p and PTable[ p ].copyset;
  PTable[ p ].access := nil;
END;
Unlock( PTable[ p ].lock );

```