Subject: Re: [PATCH 08/11] fuse: use req->page_descs[] for argpages cases
Posted by Maxim Patlasov on Thu, 25 Oct 2012 15:38:30 GMT
View Forum Message <> Reply to Message

Miklos,

> Maxim Patlasov <mpatlasov@parallels.com> writes:
>
>> @@ -888,11 +888,11 @@ static int fuse_copy_pages(struct fuse_copy_state *cs, unsigned nbytes,
>>  {
>>   unsigned i;
>>   struct fuse_req *req = cs->req;
>> - unsigned offset = req->page_descs[0].offset;
>> - unsigned count = min(nbytes, (unsigned) PAGE_SIZE - offset);
>>
>>   for (i = 0; i < req->num_pages && (nbytes || zeroing); i++) {
>>    int err;
>> +  unsigned offset = req->page_descs[i].offset;
>> +  unsigned count = min(nbytes, req->page_descs[i].length);
> Wouldn't it be cleaner if callers calculated the last page's .length
> value from the total number of bytes?  So this would just be
>
>  unsigned count = req->page_descs[i].length;
>
> And at the end of the function we can assert that nbytes went to exactly
> zero with a WARN_ON().
>
> But this is a change that needs careful testing, so maybe we're better
> off having that as a separate incremental patch later...

It cannot be as simple as 'unsigned count = req->page_descs[i].length' because in case of short reads 'nbytes' (coming from userspace) can be unpredictably small. Modulo you share my opinion that a caller of fuse_copy_pages() shouldn't modify req->page_descs[i].length.

As for WARN_ON(), we could probably guarantee that 'nbytes' <= capacity(req->pages[]) in WRITEs, but in READs, 'nbytes' comes from userspace and I'm not sure it's OK to clutter logs due to misbehaved userspace fuse (if we get 'nbytes' unexpectedly large).

Thanks,
Maxim